ABSTRACT

Title of Dissertation:	INTELLIGENT INTERSECTION MANAGEMENT THROUGH GRADIENT-BASED MULTI-AGENT COORDINATION OF TRAFFIC LIGHTS AND VEHICLES
	Manuel Rodriguez Doctor of Philosophy, 2021
Dissertation Directed by:	Professor Hosam Fathy Department of Mechanical Engineering

This dissertation examines the problem of coordinating two different types of actors in a vehicular traffic network system, namely: the traffic lights and the connected and automated vehicles traversing the traffic network. The work is motivated by an extensive previous literature showing that traffic network synchronization has substantial potential throughput and fuel economy benefits. The literature presents many algorithms for synchronizing the traversal of intersections by connected and automated vehicles (CAVs), as well as the synchronization of traffic lights within a given network. However, the integrated solution of these two synchronization problems remains relatively unexplored. The main challenge of any algorithm proposed in this area consists of managing the trade-off between computational efficiency, communication requirements, and performance.

This dissertation seeks to contribute to the list of proposed coordination strategies for CAVs and smart traffic lights by formulating a decentralized framework

based on combining ideas from gradient-based multi-agent control, trajectory planning and control barrier functions. The overall proposed control framework consists of describing vehicles and traffic lights by an extra state that directly or indirectly represent its timing (i.e arrival time for the vehicles, and switching time for the traffic lights). This timing variable evolves according to a networked multi-agent system, where the planned timing of neighboring agents governs the evolution of the planned timing of the ego agent. The planned timing state is then translated into a control action for the agents (i.e. acceleration for the vehicles, switching actuation for the traffic lights), through trajectory planning and safety regulation.

The proposed coordination framework (i) can coordinate both vehicles and traffic lights, (ii) scales efficiently to large numbers of vehicles and intersections, (iii) is computationally efficient, (iv) can work under different levels of connectivity assumptions and in the presence of human drivers, and (v) can allow for different types of coordination strategies encoded in the underlying ETFs.

INTELLIGENT INTERSECTION MANAGEMENT THROUGH GRADIENT-BASED MULTI-AGENT COORDINATION OF TRAFFIC LIGHTS AND VEHICLES

by

Manuel Rodriguez

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2021

Advisory Committee:
Professor Hosam Fathy, Chair/Advisor
Professor Derek Paley, Dean's Representative
Professor Shapour Azarm
Professor Yancy Diaz-Mercado
Professor Jin-Oh Hahn

Dedication

I would like to dedicate this dissertation to all of us who have been irritably stuck in traffic for too long.

Things will get better.

Acknowledgments

The work I present in this dissertation would not have been possible without the help and support of family, friends, colleagues and mentors.

First, I would like thank my family. I was raised in a house where inquiry was always encouraged, and curiosity was always fostered. My parents, my sister, my uncles, and my cousins are inspiring people. In their own ways, and through their own achievements, they have always inspired me not only to do meaningful work, but to find deep meaning in the work that I do. They have sacrificed much to support me in this adventure that sent me across the sea to pursue my studies. I am forever grateful to them.

I am also grateful to the wonderful friends that have walked beside me through this journey. The companionship they have offered has always been a welcome respite from the often stressful worries of doing a PhD.

I would also like to thank my committee members for their insightful feedback. A lot of the ideas in this document originated in Dr. Diaz-Mercado's class, so I feel privileged and grateful to have him in my committee.

Finally, I would like to thank my adviser, Dr. Hosam Fathy. Dr. Fathy is an outstanding engineering professor and scholar. Through his teachings I developed invaluable technical skills and sharp academic instincts. The process of doing creative research, and transforming it into valuable publications can be daunting, but with his guidance it never felt insurmountable. I am most grateful, however, not for his technical teachings, but for his mentorship. I will take with me a lot his lessons on life and leadership. He might not know he taught me these things, but they were there to learn by simply looking.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

BSFC	Brake Specific Fuel Consumption
CAV	Connected and Autonomous Vehicles
CBF	Control Barrier Function
CV	Connected Vehicles
DTTA	Desired Time Till Arrival
DTTS	Desired Time Till Switching
ETF	Edge tension funciton
FIFO	First In First Out
IIM	Intelligent intersection management
MAS	Multi-Agent System
MPC	Model Predictive Control
QP	Quadratic Program
SPAT	Signal Phase And Timing
V2V	Vehicle-to-Vehicle
V2I	Vehicle-to-Infrastructure

List of Symbols

Chapter 2

x	position of the vehicle along arterial corridor
v	speed of the vehicle
u	acceleration of the vehicle
v_{min}, v_{max}	max. and min. speed
u_{min}, u_{max}	max. and min. acceleration
L_i	position of traffic light I along the arterial corridor
I_i	length of the intersection region
R_i	interval of time the light I will red
x_{front}	position of lead vehicle
T	optimization time horizon
$\dot{m}_f$	mass fuel rate
P_e	engine power
$P_{e,\text{idle}}$	engine power at idle
$P_{e,max}$	max. engine power
η	lumped powertrain efficiency
F_prop	propulsive force
c_d	drag coefficient
A_f	frontal area of the vehicle
μ	rolling resistance
β	road grade
α	multi-objective secularization parameter
h	time step
v^a	Gipps free flow velocity model
v^b	Gipps breaking velocity model
v_n^d	desired speed of vehicle n
D_{min}	safe inter-vehicle distance at 0 speed
X_0,V_0,U_0	set of discretized states and inputs
X_k,V_k,U_k	set of constrained, discretized states and inputs
x_k,v_k,u_k	discretized states and inputs at time step k
$\hat{x_k}$,$\hat{v_k}$	intermediate state, half a time step ahead

Chapter 3

g_p	position-to-phase mapping in road segment p

λ_p	wavelength in road segment p
ϕ_p	offset in road segment p
θ_i	phase of vehicle i
s_i^d	desired distance to the intersection of vehicle i
v_i^d	desired speed of vehicle i
ω_i	natural frequency of vehicle i
ω	natural frequency shared by all vehicles in the network
K_{ij}	coupling term between agents i and j in the Kuramoto model
K	coupling term between agents in the uniform Kuramoto model
r	coherence in the order parameter
Φ	mean phase of the system
Φ_i	projection of the mean phase closest to the phase of agent i
$v_{n,p}$	nominal velocity in road segment p
s_i	distance to the intersection of vehicle i
v_i	speed of vehicle i
a_i	acceleration of vehicle i
u_i	jerk of vehicle i
$v_{i,\min}$, $v_{i,\max}$	max. and min. speed of vehicle i
$a_{i,\min}$, $a_{i,\max}$	max. and min. acceleration of vehicle i
s_i^*,v_i^*,a_i^*	optimal state trajectories
u^*	optimal input trajectory
$c_{1,\ldots,6}$	integration constants
q	max. possible flow
G	safety gap
S	safety distance

Chapter 4

h	time step
$\mathcal{L}$	set of traffic light switching events
$\mathcal{V}$	set of vehicle agents
τ_i	desired time till switching or arrival time of agent i
$\tau_{i,\min}$, $\tau_{i,\max}$	min. and max. possible arrival or switching times of agent i
u_i	coordination input
T_i	interval of possible switching or arrival time of agent i
T_H	planning time horizon
s_i	distance to the intersection of vehicle i
v_i	speed of vehicle i
a_i	acceleration of vehicle i
$v_{\min}$, $v_{\max}$	max. and min. speed
$a_{\min}$, $a_{\max}$	max. and min. acceleration

$a_{i,ff}$	free flow Gips acceleration of vehicle i
v_{a_i}	Gipps free flow velocity model
v_{b_i}	Gipps breaking velocity model
V_{ij}	edge-tension function between agents i and j
κ_{ij},$\kappa_{1,\ldots,8}$	rate parameter of logistic function
$\mathcal{N}_i$	set of neighbors of agent i
$\mathcal{G}_i$	set of traffic light switches to green for agent i
$\Delta\tau_\alpha$	interaction interval
α	interaction interval parameter
H	Hamiltonian
λ_i^s, λ_i^v	co-states
$c_{1,\ldots,4}$	integration constants
$G_{min,p}$	minimum green span window

Chapter 5

s_i	distance to the intersection of vehicle i
v_i	speed of vehicle i
a_i	acceleration of vehicle i
$v_{\min}$, $v_{\max}$	max. and min. speed
$a_{\min}$, $a_{\max}$	max. and min. acceleration
v_{des}	nominal desired velocity on the road
τ_i	desired time till switching or arrival time of agent i
u_i	coordination input
$\mathcal{L}$	set of traffic light switching events
$\mathcal{V}$	set of vehicle agents
V_{ij}	edge-tension function between agents i and j
K_{ij}	amplitude of logistic function between agents i and j
h,h_1,...,h_{10}	control barrier functions
L_f,L_g	Lie brackets of the control affine system
κ	extended class $\mathcal{K}_\infty$ function
$\gamma_{i,q}$	parameter of the q^{th} extended class $\mathcal{K}_\infty$ function, for agent i
ρ	performance function
β	nominal controller
v_i^*,a_i^*	nominal optimal speed and acceleration trajectories
u_i^*	nominal coordinating control input for agent i
$\mathcal{C}$	safe set
$U_{i,q}$	set of allowable inputs induced by CBF q, for agent i
$U_{i,q}$	set of allowable inputs for agent i
$A^{(i)}$, $b^(i)$	matrix and vector encoding the linear constraints on the input for agent i
D_{min}	safe inter-vehicle distance at 0 speed

$l(i)$	index of vehicle ahead of vehicle i
$v_{i,T}$	transition speed from one braking controller to the next for vehicle i
G	indicator function for whether a vehicle is currently planning to arrive on green
R	relaxation parameter

Chapter 1: Introduction

This dissertation examines the problem of coordinating connected and autonomous vehicles plus smart traffic lights to improve the efficiency of urban traffic intersections. Specifically, we make use of decentralized multi-agent gradient-based methods for the coordination of both vehicle arrival times and traffic light switching times, across multiple intersections. The timing coordination is coupled with an optimal control strategy that generates fuel efficient and safe speed trajectories onboard vehicle agents. Finally, adequate safety and feasility constraints are handled by a control barrier function based controller.

We call the problem of coordinating smart traffic network agents the Intelligent Intersection Management (IIM) problem. In this section, we motivate the importance of studying this problem, we review the relevant literature on this topic, and we list the contributions of our work to this body of literature.

1.1 Motivation

Congestion and inefficiencies in transportation networks are significant contributors to wasted fuel and wasted time. According to the Urban Mobility Report [1], in 2019 U.S. motorists consumed an additional 8.8 billion gallons of fuel and spent an extra 3.2 billion hours stuck in traffic. Wasted fuel and time, in turn, are detrimental to our environment and economy through their impact on emissions and productivity. Significant research on the use of connectivity and automation by vehicles and traffic infrastructure shows the potential these technologies have of improving the efficiency of traffic networks, specifically in urban scenarios. Indeed, in [2], the authors anticipate that the type of technologies presented in this section can yield at least between 10 and 20 % savings in energy. The proposed approaches work by reducing stop-and-go driving at traffic intersections through changing variables such as traffic light signal phase and timing (SPAT) and vehicle speed trajectories.

1.2 Literature Review

There is a rich body of literature that focuses on solving the IIM problem. Different approaches are proposed, and they can be categorized into different bodies of work based on which agents are considered to be controlled, namely, the traffic lights, the vehicles, or both. Approaches also differ in their formulations of the problem, and the methods used to determine vehicle and signal control. In the following sub-sections, we review some of the IIM literature by focusing on which agents are considered to be controlled, and when relevant, by focusing on approaches that make use of multi-agent control ideas in general, and gradient-based multi-agent control specifically.

1.2.1 Traffic Light Coordination

First, we consider approaches that control the signal phase and timing (SPAT) of adjacent traffic lights. The objective is to coordinate traffic lights so that their offsets give rise to “green waves” where vehicles encounter green lights in sequence, effectively avoiding start-stop behaviour. Another objective is to adapt to traffic demands as they change throughout the day and across roads. Currently, through loop detection, traffic lights can adapt their timing using strategies such as SCOOT [3] and SCAAT [4]. In [5], approaches focus on improving traffic light timing within a single intersection based on microscopic traffic data gathered from loop detectors. Most of the literature, however, focuses on the more complex problem of controlling multiple traffic lights when considering the constraints imposed by their coupled traffic flows. This problem has been extensively studied from different perspectives, and approaches based on reinforcement learning [6], fuzzy logic [7], optimal control [8, 9], and game theory [10, 11] among others have been explored.

Of particular relevance to the work proposed in this document are approaches that tackle the problem of controlling connected traffic lights through multi-agent control methods. In [12–15] traffic lights are modeled as coupled oscillators, and the Kuramoto equation [16] is used to achieve synchronization between intersections. As we show in later sections of this proposal, the Kuramoto model can be thought of as a gradient-based multi-agent control strategy. In [17], Q-learning rather than potential function shaping is used in a multi-agent control framework.

1.2.2 Connected and Autonomous Vehicle Coordination

Second, there is relevant work that focuses on the coordination of connected and autonomous vehicles (CAVs) as the controlled agents.

Some of this work focuses on vehicles adapting to fixed traffic light timing information received from upcoming lights. In [18–20], the timing of the traffic lights and the behaviour of preceding traffic are turned into constraints of a speed trajectory optimization problem. The minimization objectives can include acceleration, speed variations, and fuel consumption. The resulting behaviour is that of vehicles softly decreasing or increasing their speeds to catch green lights. In [21,22], these approaches are extended to consider the impact of mixed traffic in the overall performance of the network.

Moving on to an assumption of more connectivity and autonomy of the fleet, most of the work in CAV coordination assumes that traffic lights are not required at the intersections. Vehicles themselves, or centralized intersection planners, can coordinate safe crossings. In [23–25] extensive reviews of approaches that consider CAVs exclusively are presented. Several proposed approaches use a similar architecture as the one we put forth in this dissertation, namely, a hierarchical structure where the upper layer decides vehicles' crossing times, and the lower layer decides vehicle speed trajectories. Approaches of this type differ in how arrival times are negotiated and how speed trajectories are computed. Reservation-based approaches rely on a centralized controller that handles requests from incoming vehicles and decides on their arrival time through heuristic rules [26], optimization [27], or both [28]. In [29], the intersection coordinators are connected across multiple junctions using multi-agent consensus to improve the performance of the strategy at a network-wide level. Decentralized approaches negotiate arrival times using different methods. For example, in [30] vehicles decide their arrival time through heuristic rules like maintaining first-in-first-out (FIFO) queue orders and communicating their expected intersection exit time and speed. In [31, 32], arrival times are determined through a recursive decentralized minimization of inter-vehicles time gaps, also under a FIFO constraint.

Other approaches also use trajectory planning, but instead of computing exact arrival times for the vehicles, they include collision or priority constraints in a receding horizon optimization framework [33–36]. Notably, in [34, 35], a consensus-based multi-agent algorithm is used to negotiate priorities. In [37–39], the authors develop a collision avoidance supervisor based on the forward invariance of a safe-set; the approach is shown to indirectly solve the scheduling problem for simple traffic scenarios.

Another type of all-CAV approach makes use of the idea of virtual platoons, where all vehicles approaching an intersection are mapped into a single virtual road, and a car following controller tracks a desired headway [40–42]. Naturally, these approaches preserve FIFO schedules. Interestingly, in [41, 42], the car following controller makes use of gradient-based multi-agent methods to track inter-vehicle distances.

1.2.3 Traffic Light and CAV Coordination

Finally, we can look at the problem of coordinating both traffic lights and vehicles together, which is most relevant to the future contributions presented in this proposal. In the literature, this problem is mostly tackled from the perspective of optimization [43–48]. Proposed approaches generally divide the problem into a vehicle arrival time and traffic light switching time optimization layer, coupled with a vehicle speed trajectory optimization layer. The explored methodologies differ in their chosen traffic light and vehicle models, their optimization objectives, and the algorithms used to solve the problem. Table 1.1 summarizes the assumptions and solution approaches considered by several relevant references.

Reference	Assumptions	Approach
[47]	· Single intersection controlled, with multiple lanes, and no overtaking. · Predetermined signal phase groups. · Full vehicle autonomy and connectivity	· Rolling horizon centralized travel time minimization of switching and arrival times, coupled with onboard fuel consumption minimization of vehicle speed trajectory. · Timing optimization is achieved by coarse discretization and brute force methods. · Speed trajectory optimization is done using pseudo-spectral methods
[49]	· Single Intersection controlled, with multiple lanes, and no overtaking. · Predetermined Signal Phase groups. · Mixed traffic (CAVs and human-driven vehicles)	· Rolling horizon centralized co-optimization of switching and arrival times, coupled with speed trajectory optimization · Optimization is done using a travel time minimizing dynamic programming algorithm, with a shooting heuristic sub-routine for the optimization of vehicle trajectories
[46]	· Cell transmission macroscopic traffic model with coarse discretization (6 second time step, 80 m long cells). · Large network of intersections. · Full vehicle autonomy and connectivity	· Maximization of cumulative throughput and minimization speed variations. · The network level MILP is decomposed into intersection level sub-problems coupled through relaxed interdependent constraints between neighboring intersections.

Table 1.1: CAV and Traffic Light Coordination Literature

1.2.4 Key Lessons

From the above literature, we can learn the following key lessons. First, significant fuel and throughput improvements can be achieved from leveraging connectivity and automation of the two main agents in a traffic network: traffic lights and vehicles. Second, the intelligent intersection management problem has different temporal and spatial scales that approaches need to consider. Temporally, control strategies need to adapt to traffic demands that change throughout the day. Spatially, control strategies need to consider the impact of neighboring intersections. Third, the CAV speed trajectory optimization problem with a given arrival time is well understood for several optimization objectives (acceleration, fuel consumption), and for some of these, efficient solution methods are proposed.

1.3 Open Challenges and Research Contributions

The IIM literature shows the great potential connectivity and automation have of improving traffic networks. However, before true implementation of these technologies can be achieved, key open challenges still need to be addressed. While there are many gaps in this literature, here, we focus on the ones that motivate the contributions of our work. In the following, we identify these open questions and the corresponding contributions of our work.

First, most of the work cited above focuses on the control of passenger vehicles. Heavier vehicles suffer from more restrictive power constraints, which make the optimization of their speed trajectory a more challenging problem. In **Chapter**

2 we explore the benefits of fuel consumption minimization for a heavy-duty truck traversing an arterial corridor with known SPAT information of the upcoming traffic lights.

Second, coordination between CAVs themselves, without traffic lights is usually restricted to interactions within single intersections. Multi-intersection approaches usually rely on centralized coordinators that handle the coupling between intersections. As such, most approaches do not explore how network wide coordination can be achieved through decentralized CAV negotiation. In **Chapter 3** we address this gap by proposing a synchronization strategy for CAVs travelling within urban networks based on a non-linear multi-agent consensus equation known as the Kuramoto equation. The approach models vehicles as travelling waves defined by their phases and frequencies, which are related to their desired positions and velocities. Through the Kuramoto equation, vehicles synchronize their phases, and the careful design of the mapping between position and phase guarantees that synchronized phases correspond to efficient intersection crossings.

Third, the problem of coordinating both traffic lights and vehicles is relatively unexplored [49]. Specifically, approaches that consider both agents together rely on optimization methods to determine the timing of traffic light switches and vehicle arrivals. While optimizing for desired objectives might yield the most improvements in those objectives, optimization methods can suffer from being computationally complex, centralized, and difficult to scale. In **Chapter 4** we address this challenge by proposing a gradient-based multi-agent framework for the joint coordination of traffic light switching times and vehicle arrival times. The method uses logistic functions to characterize the potential between two agents whether they are traffic lights or vehicles. This potential is given by an edge-tension function (ETF). The strategy is defined and evaluated for single intersections. We note at the end of the chapter how this framework is a generalization of the strategy presented in the previous chapter, given that the Kuramoto governed interaction between agents can be described through suitable ETFs.

Fourth, the above contributions focus mostly on the problem of coordinating the timing of the agents. Safety constraints are satisfied by either predictive planning, which requires strong connectivity and computational capabilities, or ad-hoc imposition of constraints, which suffers from weak mathematical guarantees. In **Chapter 5** we introduce a safety regulator that makes use of theoretical tools of non-linear controls, namely, control barrier functions. Safety and feasibility constraints are formulated for both the traffic lights and vehicles, and using the control barrier functions associated with these constraints, a controller that modifies the nominal coordinating strategy is synthesized.

Chapter 2: Speed Trajectory Optimization for a Heavy-Duty Truck Traversing Multiple Signalized Intersections: A Dynamic Programming Study

2.1 Introduction

[1] As a motivating starting point, in this chapter we explore the fuel savings that can be achieved by optimizing the speed trajectory of a heavy-duty truck traversing a sequence of intersections, under the assumptions that the behavior of the leading traffic and the timing of the traffic lights is known. Specifically, we look at the impact of corridor topology (i.e. green cycle lengths, phase offsets) on the expected fuel savings of the optimized trajectories. This is an important area of research because vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) technology has the potential to allow autonomous vehicles to reduce fuel consumption, especially in urban and sub-urban driving scenarios. The literature tackles the problem of arterial corridor trajectory optimization, and shows the potential fuel saving benefits. However, previous research focuses primarily on passenger vehicles, and often limits its findings to specific case studies. The main contribution of this chapter is to offer

[1] The work presented in this chapter is adapted from [50], accepted in the *2018 IEEE Conference on Control Technology and Applications (CCTA)*

an estimate of the fuel saving potential – for heavy-duty trucks and under different corridor characteristics – of optimizing trajectories in an urban arterial with V2V and V2I capabilities.

Several researchers have proposed different formulations and optimization techniques to find a speed trajectory that improves fuel economy for passenger vehicles traveling through arterial corridors when traffic light timing information is available. Mandava et al. explores the performance of an algorithm that minimizes acceleration maneuvers for a single vehicle with no leading traffic [18] . In [51], Asadi and Vahidi propose a hierarchical control structure where a higher level algorithm feeds constraints that reduce idling at red lights to a model predictive controller (MPC) that minimizes acceleration and avoids collisions. This strategy is expanded in [19] to explicitly optimize for fuel efficiency at the MPC layer, and it is further expanded in [22] to explore the impact of coordination between connected vehicles (CV) on the fuel savings of the entire fleet for different levels of CV penetration. In [20], He et al. optimize the speed trajectory for fuel economy with an added constraint generated by a predicted queue length. In terms of fuel consumption these different approaches show savings ranging from about 12% when minimizing acceleration [18] to up to 40% when maximizing fuel efficiency [20] in their particular case studies.

As an optimal control problem, minimizing fuel consumption under the presence of traffic light constraints is a highly non-convex problem due to the form of the cost and the constraints. Fuel consumption is a function of power, which is a bilinear function of inputs and states. Moreover, the position constraints due to traffic lights are time-dependent and highly discontinuous. For this reason, the previously highlighted literature makes use of heuristic rules and simplifying assumptions in order to develop algorithms capable of generating online solutions. Their findings, while promising, focus mainly on passenger vehicles and generally do not solve for global optimality.

In this chapter, we consider heavy-duty trucks as opposed to passenger vehicles, and we use dynamic programming to find the trajectories that maximize fuel economy in arterial corridors with different traffic conditions, different light timing cycles, and synchronization offsets. The solutions are compared to a Gipps model of human car following behavior, and the fuel savings are related to macroscopic performance measures.

2.2 Problem Formulation

We consider a string of vehicles approaching a sequence of traffic lights. The last vehicle in the string is assumed to be an autonomous heavy-duty truck, whose trajectory is given by the solution of the optimization problem presented in part A. of this section. The vehicles in the convoy ahead of it follow a car-following driving pattern governed by a Gipps model modified to handle traffic signals as explained in part B.

2.2.1 Optimization Problem

In this section we formulate the optimization problem whose solution determines the trajectory of the heavy-duty truck. For a supervisory, motion planning algorithm it is reasonable to use the speed v and the position x as the state variables of the system, and the acceleration u as the control input variable. The position is initialized a distance L_i from each traffic light i in the sequence, and we assume the vehicle receives information on the interval of time R_i that traffic light i is red. We also assume that the position $x_{front}(t)$ of the vehicle ahead is known, either by predicting it or receiving it through vehicle-to-vehicle connectivity.

Since we will be minimizing for fuel consumption, we begin by developing a method of computing the mass fuel rate of the truck from its state and input variables. In this chapter, we will use an empirical brake specific fuel consumption (bsfc) map taken from a proprietary model of a Volvo truck with a diesel engine. For simplicity, and to keep the number of state variables and inputs to a minimum, we assume the truck's powertrain has a continuously varying transmission and that the controller can operate the engine at its optimal brake specific fuel consumption point, , for any demanded engine power . The mass fuel rate will then be given by:

$$\dot{m_f}(t) = P_e(t)\mathrm{bsfc}^*(P_e(t)) \tag{2.1}$$

Where the engine power is given by the product of the propulsive force at the wheels, the speed of the vehicle, and the lumped powertrain efficiency.

$$P_e = \begin{cases} \eta F_{prop} v & F_{prop} > 0 \\ P_{e,\mathrm{idle}} & F_{prop} \leq 0 \end{cases} \tag{2.2}$$

The propulsive force, in turn, is given by the longitudinal force balance on the vehicle, which includes aerodynamic drag, rolling resistance and weight:

$$\begin{aligned} F_{prop} = mu + \frac{1}{2}c_d A_f v^2 \\ + mg\mu\cos(\beta(x)) + mg\sin(\beta(x)) \end{aligned} \tag{2.3}$$

where m, c_d, A_f and μ are the mass, drag coefficient, frontal area and rolling resistance coefficient of the truck. The variable β is the road grade, which is a function of position. In this study, however, we assume a flat road:

$$F_{prop} = mu + \frac{1}{2}c_d A_f v^2 + mg\mu \tag{2.4}$$

With a model for $\dot{m}_f$, we are ready to state our optimization problem. Given that simply minimizing fuel consumption can lead to the trivial solution of having the vehicle stop and idle, most algorithms attempt to minimize fuel consumption per distance traveled instead. Rigorously doing so implies that we minimize the ratio of two integrals: the integral of mass fuel rate over the integral of velocity. This is usually avoided by simply minimizing the integral of the ratio of fuel rate over speed. In the interest of exploring the inherent trade-offs between minimizing fuel consumed and maximizing distance traveled, in this chapter we formulate the optimization objective as a linear combination of fuel rate and velocity. Using α as our Pareto weight, we can express this multi-objective optimization problem as follows:

$$\min_{x,v,u} \int_0^T [(1-\alpha)\dot{m_f} - \alpha v]\mathrm{dt} \tag{2.5a}$$

Subject to:

$$\dot{x}(t) = v(t) \tag{2.5b}$$

$$\dot{v}(t) = u(t) \tag{2.5c}$$

$$x(t) \leq x_{front}(t) - D_{\min} \tag{2.5d}$$

$$x(t) \notin \{L_i \leq x(t) \leq L_i + I_i | t \in R_i\} \tag{2.5e}$$

$$0 \leq v(t) \leq v_{max} \tag{2.5f}$$

$$u_{min} \leq u(t) \leq u_{max} \tag{2.5g}$$

$$P_{e,idle} \leq P_e(t) \leq P_{e,max} \tag{2.5h}$$

The first and second term in the cost function (2.5a), when integrated, give the fuel consumed and distance traveled, respectively. The constraints (2.5b) and (2.5c) are the dynamics of the vehicle, which amount to a double integrator when we use acceleration as the input variable. Constraint (2.5d) guarantees that the truck keeps a safe distance from the vehicle in front. The next constraint, (2.5e), states that the vehicle cannot be within intersection i when the light is red; L_i and I_i are the location and size of the intersection, and R_i is the set of times the intersection light is red. The last three constraints are the bounds on the feasible velocities and inputs; those are given speed limits of the road, drivability considerations, and power limits of the engine.

2.2.2 Car-Following Model

To model the human-driven vehicles that precede the truck, we follow the example of [22], and use a Gipps driving model, modified to account for traffic lights. The Gipps model is an empirical car following model parametrized by the desired cruising speed, and the maximum acceleration and braking drivers are willing to undertake [52]. Given the speed v_n of vehicle n at time t, the speed at an instant $t+h$, where h is a time step in the order of the reaction time of the driver, is taken to be the smallest of two limits: $v^a(t+h)$, and $v^b(t+h)$.

$$v_n(t+h) = \min[v^a(t+h), v^b(t+h), v_{n,max}(t+h)] \tag{2.6}$$

The first term governs the behavior when the preceding vehicle is not within close proximity, and it guarantees the vehicle tracks its desired cruising speed. It is given by:

$$\begin{aligned} v^a(t+h) =& v_n(t) \\ & + 2.5u_{n,max}h(1-\frac{v_n(t)}{v_n^d})\sqrt{0.025+\frac{v_n(t)}{v_n^d}} \end{aligned} \tag{2.7}$$

The second terms governs the braking behaviour that drivers undertake to keep a safe distance from the vehicle ahead, and it is given by:

$$v^b(t+h) = u_{min}h + \left[u_{n,min}^2 h^2 - u_{min}\left(2(x_{n-1}(t) - D_{min} - x_n(t)) - v_n(t)h - \frac{v_{n-1}^2}{u_{n-1,min}}\right)\right]^{\frac{1}{2}} \tag{2.8}$$

where $u_{n,max}$, $u_{n,min}$, v_n^d are the maximum acceleration and maximum braking, and the desired velocity of vehicle n, which will be set as the speed limit of the road. D_{min} is the safety distance kept by the driver, which is be the same as the one in eq. (2.5). When the vehicle is the leading vehicle, or there are no vehicles between itself and the next intersection, we set the position of the preceding vehicle x_{n-1} to be the position of the next light that is red. In this situation, the speed v_{n-1} is set to be 0.

Finally, the last term in eq. (2.6) is a modification from the original Gipps model, and it was added to ensure the vehicle does not undertake an acceleration that results in an unfeasible engine power demand.

2.3 Dynamic Programming Solver

In this section we go through the choices we made in transforming the continuous time optimization problem given by eq. (2.5), into a discrete time problem suitable to be solved using the standard dynamic programming approach. We also look at the Pareto trade-off of minimizing fuel consumption versus maximizing distance travelled to select our scalarization weight α, and how this compares to maximizing fuel efficiency.

2.3.1 Discretization

We define a time step h, which gives $N = T/h$ total time instants to optimize for. The discrete form of the underlying problem of eq. (2.5) is then:

$$\min_{x_k, v_k, u_k} \sum_{k=1}^{N} [(1-\alpha)\dot{m}_f(\hat{x_k}, \hat{v_k}, u_k) - \alpha \hat{v_k}]h \tag{2.9a}$$

Subject to:

$$x_{k+1} = x_k + v_k h \tag{2.9b}$$

$$v_{k+1} = v_k + u_k h \tag{2.9c}$$

$$x_k \in X_k \tag{2.9d}$$

$$v_k \in V_k \tag{2.9e}$$

$$u_k \in U_k \tag{2.9f}$$

where $\hat{x_k}$ and $\hat{v_k}$ are intermediate states; that is:

$$\hat{x_k} = x_k + v_k \frac{h}{2} \tag{2.10a}$$

$$\hat{v_k} = v_k + u_k \frac{h}{2} \tag{2.10b}$$

In using $\hat{x_k}$ and $\hat{v_k}$ to compute $\dot{m_f}$ in eq. (2.9a), we are effectively using a midpoint rule to approximate the integration. Constraints (2.9b) and (2.9c) are obtained by approximating the time derivatives using a finite difference. Finally, constraints (2.9d - 2.9f) lump equations (2.5d - 2.5h) into a single, time dependent, set constraint for each state and input variable. X_k, V_k, and U_k, are the intersections between the state and input meshes X_0, V_0, and U_0 and the sets defined by (2.5d - 2.5h), and they are computed at every time step when the dynamic programming algorithm runs.

The state meshes X_0, V_0 are chosen to be consistent with the time step h:

$$X_0 = \{x = n\Delta x \,|\, x \leqslant \max(x_{front}(t)) \,\&\, n \in \mathbb{N}\} \tag{2.11a}$$

$$V_0 = \{v = n\Delta v \,|\, v \leqslant v_{\max} \,\&\, n \in \mathbb{N}\} \tag{2.11b}$$

$$U_0 = \{u = u_{\min} + n\Delta u \,|\, u \leqslant u_{\max} \,\&\, n \in \mathbb{N}\} \tag{2.11c}$$

where the discretization steps must satisfy:

$$\Delta v = \frac{\Delta x}{h} \tag{2.12a}$$

$$\Delta u = \frac{\Delta v}{h} \tag{2.12b}$$

Vehicle Parameters		Optimization Parameters	
m	35000 kg	Δx	0.5 m
c_d	0.49	Δv	0.5 m/s
A_f	10.67 m^2	Δu	0.5 m/s^2
μ	0.01	h	1 s
β	0 rad	D_{min}	10 m
η	0.90	α	0.3

Table 2.1: Vehicle and Optimization Parameters

To allow for a large number of inputs without extensively compromising computational time, we select Δx and h to be 0.5 and 1 respectively. It follows that Δv and Δu are both 0.5 as well. The rest of the parameters used in subsequent simulations is summarized in Table 2.1.

2.3.2 Weight Selection

At this point, the only thing we need to define before running simulations is the scalarization weight α, which lies between 0 and 1. When α is 0, we focus only in minimizing fuel consumption, which will be achieved by not moving at all and keeping the engine at idle. When α is 1, we only focus on maximizing range; at this value, the problem is ill-defined since any solution that reaches the end of the position state mesh would have the same cost. We will select a sample corridor and compute the optimal solution for different values of α.

For this section, we look at a 1.2 km stretch of road representing College Av., State College. This corridor contains 6 signalized intersections, whose timing, location and dimensions were estimated using live measurements and map data. A map view of the road in question, along with the relevant measurements is shown in Figure 2.1. We consider the scenario where the truck enters the control zone following a string of 5 vehicles modeled using equation (2.6) and the parameters given in Table I. We use the values given in Table I for the parameters of the truck, and solve the optimization problem given by (2.5) using DP, for different values of α and a time horizon T equal to the time it takes the preceding vehicle to reach the last light.

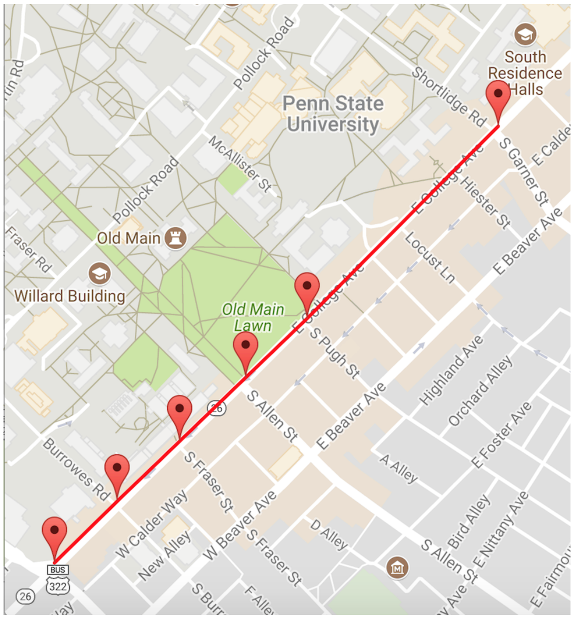
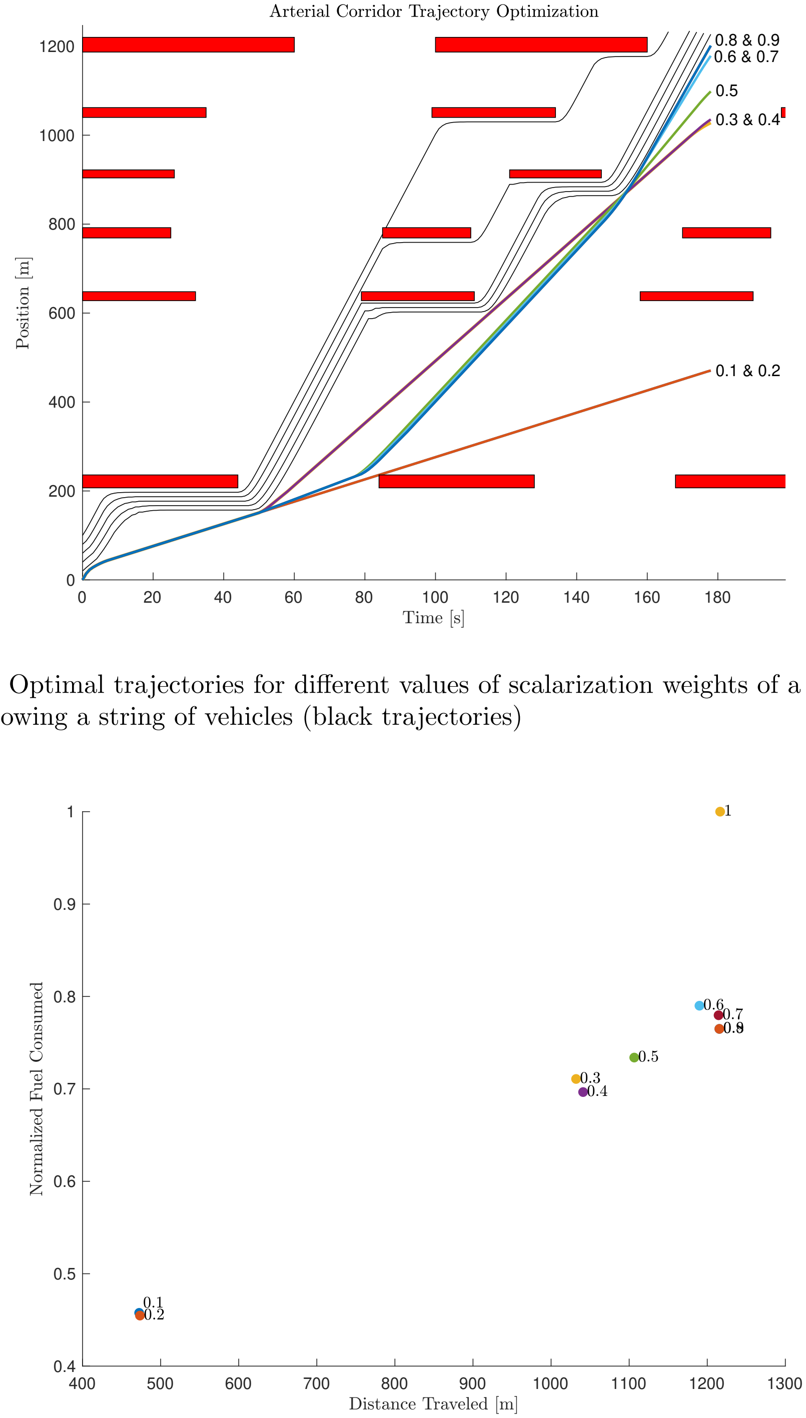


Figure 2.1: Case Study: 6 intersection corridor in College Avenue, State College, PA

Figure 2.2(a) shows the optimal trajectory of the truck for each value of α in a space-time diagram. The red rectangles represent the times the lights are red, and the black lines represent the trajectories of human-driven vehicles. Figure 2.2(b) shows the associated fuel consumption and distance traveled for the multi-objective optimization problem at hand. We can see a sharp jump in both range and fuel consumed between $\alpha = 0.2$ and $\alpha = 0.3$ after which a change in α does not affect the cost, or the shape of the solution as much.



(a) Optimal trajectories for different values of scalarization weights of a truck following a string of vehicles (black trajectories)

(b) Costs of the multi-objective optimization problem

Figure 2.2: Trajectories and costs for minimizing fuel consumption vs maximizing range over 6 intersections in College Av.

Based on the shape of the trajectories for this specific case study, and the fact that our ultimate goal is to maximize fuel efficiency, which in terms of grams of fuel per meter is the ratio of our two objectives, we select $\alpha = 0.3$ as our weight for the rest of the simulations.

2.4 Simulation Results

In this section we assess the ability of trajectory optimization in reducing fuel consumption by running simulations for different arterial corridor configurations. The purpose is to estimate the expected fuel savings based on the available traffic light information. Specifically, we focus on the two main parameters that traffic light management systems can toggle: the green cycle length and the offset between one light and the next. A properly designed corridor will optimize the portion of the cycle that each light is green based on the volume of traffic, and it will synchronize green lights to allow traffic to flow as uniformly as possible without stopping at the intersections.

We consider a stretch of road with 4 evenly spaced intersections, and a convoy of 5 vehicles traveling in front of the truck when it enters the control zone 150 m from the first light. Each intersection is 20 m long, and is 150 m away from the previous one. Their traffic lights have a cycle length of 90 s, of which a certain ratio, the green cycle ratio, corresponds to a green light. The timing cycles of each light are offset by a certain amount of seconds with respect to the previous light. We will vary both the green cycle ratio and synchronization ratio independently, and see the amount of fuel that can be saved when compared with a truck that follows a Gipps driving model.

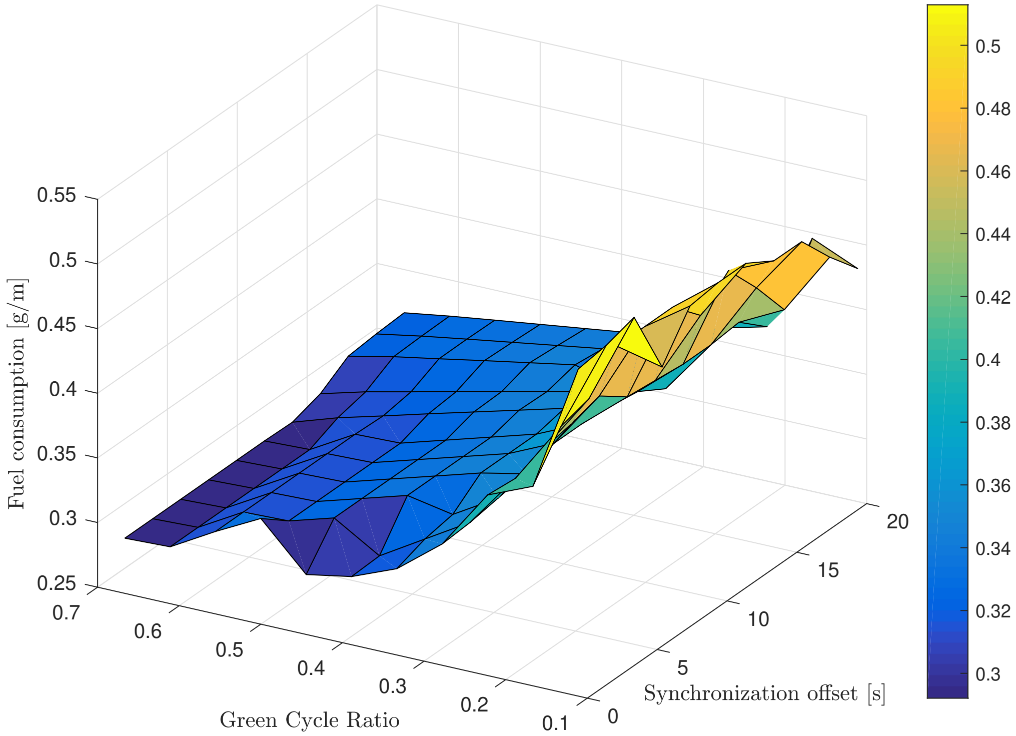


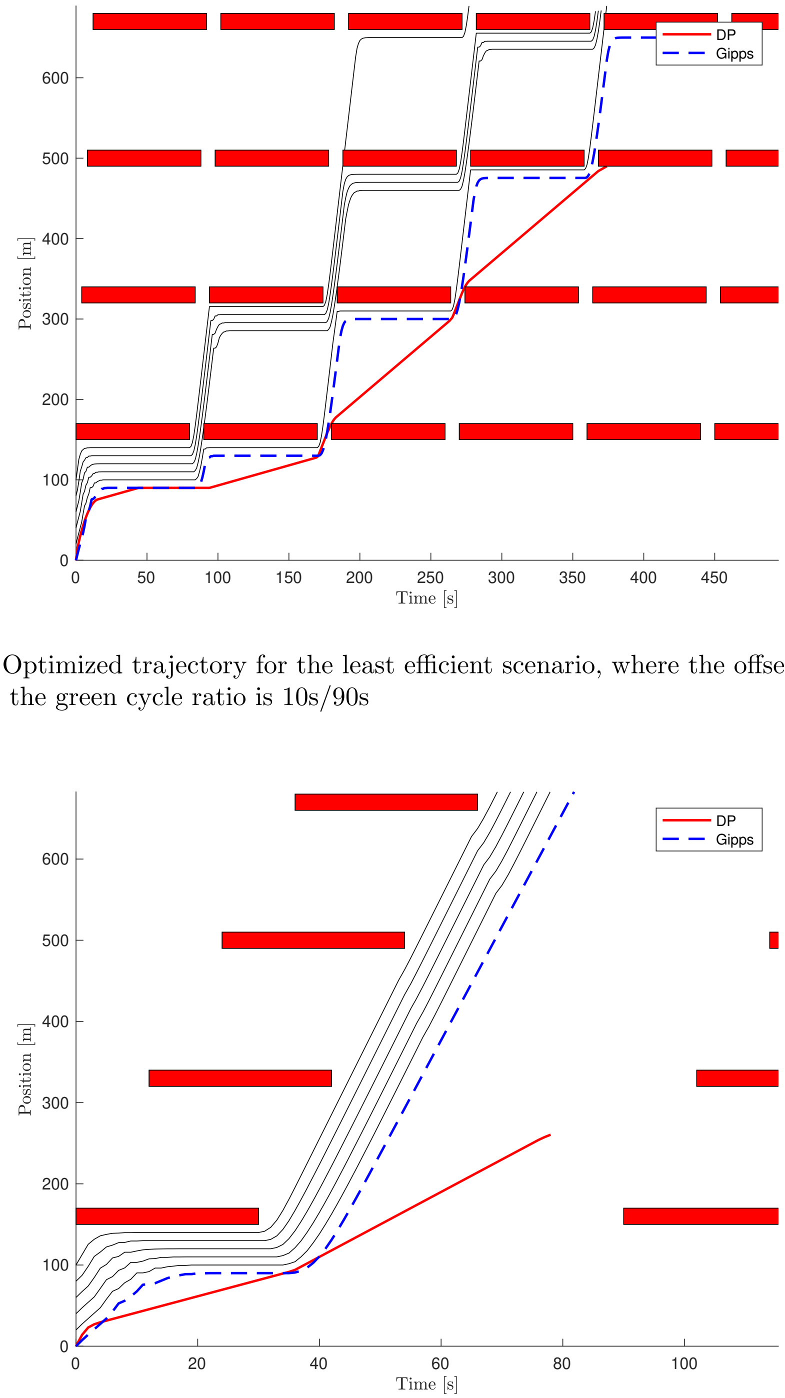
Figure 2.3: Fuel consumption in grams/m achieved by optimizing the trajectory for different green cycle ratios and synchronization offsets

We can see in Figure 2.3 the results of our simulation study. The achieved fuel consumption ranges from 0.29 to 0.51 grams/m, and the scenarios where these occur correspond, as expected, with the situation where the corridor is properly versus poorly timed and synchronized. Figures 2.4(a) and 2.4(b) show the optimized trajectories in both of these extreme cases. When green intervals are short, and the offset is slower than the speed of the convoy, the worst fuel efficiency occurs; on the flip side, when the offset matches the speed of the string of vehicles ahead, which occurs when it is equal to spacing between intersections divided by the desired velocity of the traffic, the algorithm achieves its best fuel economy. These differences, however, are the product of having an optimized corridor, rather than smart look-ahead driver.

Figure 2.5 shows, in percentages, how much fuel is saved by performing trajectory optimization when compared to a baseline car following model showed as a dotted line in Figure 2.4. We can see that the improvements range from around 34% to 72% depending on how well the traffic lights are timed and synchronized. The smallest fuel saving is achieved for an offset of 12 s and green cycle of 20s/90s. We notice that the maximum fuel savings of above 70%, which occur when the corridor is poorly timed, do not coincide with the minimum fuel consumption, which happens at long green cycle ratios and good offset synchronization.

2.5 Conclusion

In the interest of exploring the fuel saving benefits for heavy-duty trucks of trajectory optimization in the presence of traffic and traffic light information, we use dynamic programming to minimize fuel consumption. The problem is formulated as a minimization of two competing objectives, fuel consumed and distance traveled. We then explore the underlying Pareto trade-off to select an appropriate scalarization weight of 0.3. Finally, we simulate the performance of the global optimizer for different light timing parameters to find the expected fuel savings when traversing different corridors. The fuel savings are determined by comparing to human driving



(a) Optimized trajectory for the least efficient scenario, where the offset is 4 s and the green cycle ratio is 10s/90s

(b) Optimized trajectory for the most efficient scenario, where the offset is 12 s and the green cycle ratio is 60s/90s

Figure 2.4: Example trajectories for the best and worst scenarios, including leading traffic (black), human-driven truck(dotted), and DP optimized trajectory

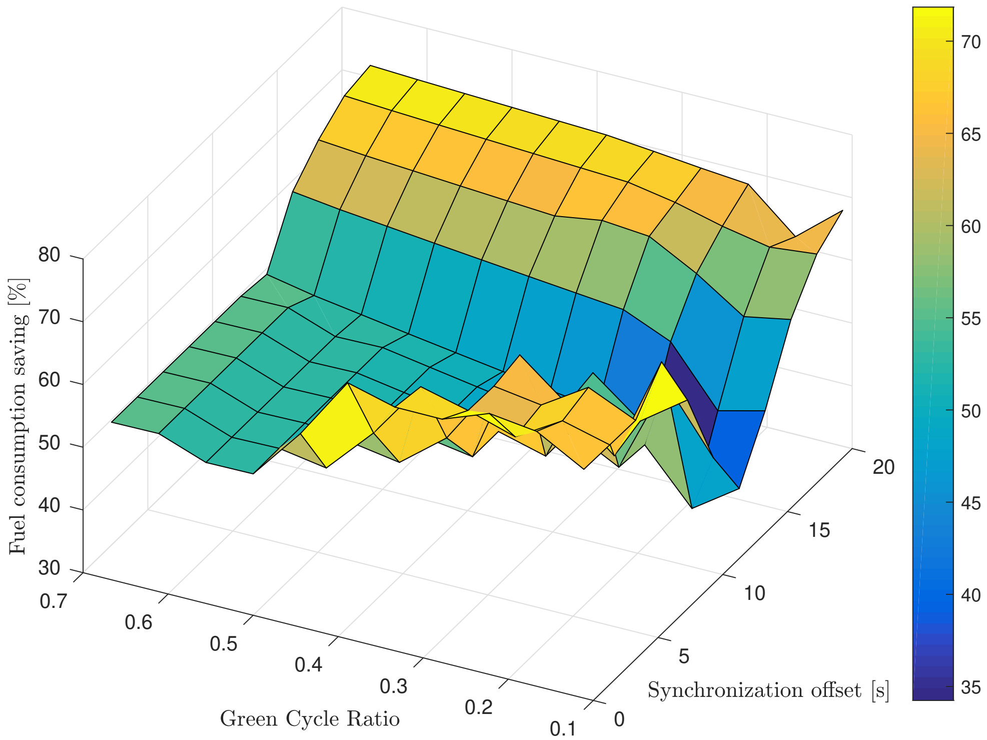


Figure 2.5: Percentage fuel consumption reduction when comparing optimized versus Gipps car-following trajectories for different green cycle ratios and synchronization offsets

behavior as modeled by a Gipps modeled.

Our results show that trajectory optimization in arterial corridors is potentially more beneficial for heavy-duty trucks than for passenger vehicles. Indeed, we find fuel savings ranging from 32 to 72%, while values for passenger vehicles reported in the literature do not usually exceed 40%.

These results motivate further efforts in developing real time speed advisory optimization algorithms for heavy-duty trucks. Furthermore, since dynamic programming solutions guarantee global optimality, they serve as an effective benchmark to which compare faster online controllers.

Chapter 3: Distributed Kuramoto Self-Synchronization of Vehicle Speed Trajectories in Traffic Networks

3.1 Introduction

3.1.1 Overview

[1] The study in the previous chapter motivates the development of coordination algorithms that reduce stop and go behaviour of vehicles as they traverse intersections. Moreover, it motivates the importance of considering the interactions between adjacent intersections. We see from the results of the dynamic programming study that the value of predictive control and trajectory planning is diminished when the lights in an arterial corridor are not properly synchronized. As we discuss in Chapter 1, synchronizing traffic lights is an active area of traffic engineering research. However, the futuristic coordination strategies that coordinate only CAVs typically do not consider multiple intersections, and are designed locally. In this chapter we consider the traditional problem of coordinating only CAVs, but we explicitly design a strategy that considers interactions between vehicles approaching neighboring intersections. The algorithm proposed here, maintains computational efficiency by

[1] The work presented in this chapter is adapted from [53,54], accepted in *2019 American Control Conference (ACC)*, and *IEEE Transactions on Intelligent Transportation Systems* respectively.

relying on a simple decentralized coordination law: the Kuramoto equation, whose computational complexity grows linearly with the number of agents [55]. We start by looking at this problem without traffic lights for the purpose of simplicity; in the next chapters, we explore how this multi-agent consensus strategy can be modified to include traffic lights as controlled agents.

Indeed, this chapter presents a distributed synchronization strategy for CAVs in traffic networks. The strategy considers vehicles traveling from one intersection to the next as waves. The phase angle and frequency of each wave map to its position and velocity, respectively. The goal is to synchronize traffic such that intersecting traffic waves are out of phase at every intersection. This ensures the safe collective navigation of intersections. Vehicles share their phase angles through the V2X infrastructure, and synchronize these angles using the Kuramoto equation. This is a classical model for the self-synchronization of coupled oscillators. The mapping between phase and location for vehicles on different roads is designed such that Kuramoto synchronization ensures safe intersection navigation. Each vehicle uses a constrained optimal control policy to achieve its desired target Kuramoto phase at the upcoming intersection. The overall outcome is a distributed traffic synchronization algorithm that simultaneously tackles two challenges traditionally addressed independently, namely: coordinating crossing at an individual intersection, and harmonizing traffic flow between adjacent intersections. Simulation studies highlight the positive impact of this strategy on fuel consumption and traffic delay time, compared to a network with traditional traffic light timing.

3.1.2 Literature review

Coordinating traffic at intersections requires solving two different problems, at two different scales. The first is servicing conflicting flows at an intersection so that vehicles do not collide with each other; in other words, deciding who crosses when. We refer to this problem as the intra-junction coordination problem. The second problem is harmonizing the flow between adjacent intersections to reduce the amount of energy vehicles waste due to frequent acceleration and braking; we refer to this problem as the inter-junction coordination problem. As we highlight in Chapter 1, most of the literature on intersection control focuses on one of these problems individually. Some approaches combine separate solutions, and evaluate their performance when combined.

Coordination between adjacent traffic lights for example, aims to solve the inter-junction problem by synchronizing the timing of the lights. Of specific relevance to the work presented in this chapter, several proposed traffic light control approaches make use of the Kuramoto equation for self-synchronizing oscillators. Sekiyama et al. proposed such an approach, where they used Kuramoto synchronization to adjust signal phase and timing [12]. This work was further expanded in [13–15]. In general the problem can be thought of as material transport problem in a directed graph, as explained by Lammer et. al. in [14].

Assuming vehicle fleet connectivity, some approaches avoid the use of traffic lights altogether by having the vehicles coordinate crossing times with each other, or with a centralized coordinator [23, 24]. In general the problem is solved in two layers. The first one determines vehicles' crossing times or sequence, while the second controls the vehicles' speeds to achieve the agreed-upon crossing time. The approach proposed in this work follows a similar structure. Other approaches based on the formation of virtual platoons are also of particular relevance to this work, because they make use of multi-agent consensus strategies, which can be thought of as linear counter-parts of the Kuramoto equation. Vaio et al. [41] propose a decentralized protocol that projects vehicles in conflicting roads into the same coordinate system, namely a distance to the upcoming intersection. Through a heuristic algorithm vehicles negotiate desired inter-vehicular distances, and they use modified consensus to achieve the desired formation. The method is evaluated for a single intersection. Most of the above approaches focus on solving the intra-junction problem at single intersections.

Some approaches have both connected centralized agents, that can communicate with each other at different intersections, and connected vehicles that exchange information with the coordinators and among themselves. These approaches can attempt to solve both the inter-junction and intra-junction problem. In [56], a centralized reservation-based controller at each intersection communicates its decisions to both the vehicles it is in charge of scheduling and to the controllers at neighboring intersections. The crossing time decisions are made by solving a mixed-integer linear program that considers the information it receives from its adjacent intersection managers. The approach is evaluated both with and without coordination between intersection managers, showing that when intersection controllers can communicate with each other, fuel consumption benefits double. A similar approach is taken in [29], where connected centralized schedulers take into account each other's information when making reservation decisions. In this case, coordination between the schedulers is achieved by the use of multi-agent consensus, as opposed to optimization.

The above literature highlights the breadth of different approaches that are proposed to solve the autonomous intersection management problem. From this literature, we identify the following key lessons. First, Kuramoto models and other consensus-based approaches have been successfully used to coordinate traffic lights and centralized intersection coordinators, but they have not been explored as means to coordinate autonomous vehicles themselves. Second, approaches that consider multiple intersections and the coupling between them can yield larger fuel savings compared to localized controls. However, most approaches that solve both intra- and inter-junction problems rely on some sort of centralized agent that couples intersections. This chapter proposes an approach to solve the autonomous intersection management problem at both levels using the non-linear consensus equation known as the Kuramoto model. The use of Kuramoto allows vehicles to first agree upon the current state of the intersections (i.e. which flow is being serviced), and then to synchronize with the intersections.

3.1.3 Outline

The rest of the chapter is organized as follows. In Section 2 the proposed strategy is presented. Section 3 discusses some of the theoretical implications of the strategy and its design parameters on traffic flow, throughput, and safety. Finally, section 4 presents simulation results to validate the proposed strategy.

3.2 Proposed Strategy

We consider a grid of interconnected intersections in an urban traffic network. We assume that all incoming vehicles are autonomous, capable of vehicle-to-vehicle (V2V) communication, and can interact with all the vehicles in the network (i.e. all-to-all connectivity). Less restrictive communication topologies are possible as shown in the Kuramoto consensus literature [57], and do not alter the fundamental ideas behind this work.

The idea behind our proposed strategy consists of mapping the position and velocity of each vehicle to a corresponding virtual phase and frequency. The vehicles exchange phase information through V2V communication, and compute the dynamics of their phases using the Kuramoto equation. This naturally drives them to synchronize. From the phase trajectories, vehicles determine the times and velocities at which they need to arrive to the upcoming intersections. With this information, they formulate a linear quadratic optimal control problem that is solved at each time step to determine the acceleration command that will place them at the intersection at the right time, with the right speed. The mapping between phase and position must satisfy certain constraints for this strategy to produce the desired behaviour, that is, safe crossing at intersections (which solves the intra-junction problem) and smooth crossing between intersections (which addresses the inter-junction problem).

In this section we describe in detail how the proposed strategy can be implemented in a network of roads with or without right turns, where all vehicles are autonomous and inter-connected.

3.2.1 Kuramoto Synchronization

The literature on traffic light synchronization using the Kuramoto equation works by describing the agents (i.e the traffic lights) as oscillators and establishing a mapping between the phase of the agent and a control action (i.e switching from green to yellow, or red). In our proposed strategy, where the vehicles are the agents as opposed to the traffic lights, the mapping relates the phase of the vehicles to a position along the road. For each road segment p, we define a mapping $g_p(\theta)$ that relates the phase θ_i of a vehicle i to the vehicle's desired distance to the intersection along the curvature of the road s_i^d:

$$s_i^d = g_p(\theta_i) \tag{3.1}$$

We choose g_p to be an affine function of phase; it can therefore be described by two parameters. We call these parameters the wavelength λ and the offset ϕ, where λ is the slope and ϕ the zero crossing. For a given road p, the mapping is then:

$$g_p(\theta_i) = (\theta_i - \phi_p)\frac{\lambda_p}{2\pi} \tag{3.2}$$

We can think of this map as having wrapped the length along the road around a circle of radius $\frac{\lambda_p}{2\pi}$, and rotated it by an angle ϕ_p

Assuming steady state tracking of the desired distance to the intersection s_i^d, it follows from the definition of g_p that a vehicle i on road p will reach the intersection when its phase θ_i is equal to the corresponding offset ϕ_p. It also follows that two vehicles on road p with a phase difference of some multiple k of 2π, will be separated by $k\lambda_p$ meters. Mathematically, these two properties of our mapping can be expressed as:

$$g_p(\phi_p) = 0 \tag{3.3}$$

$$g_p(\theta_i + 2k\pi) - g_p(\theta_i) = k\lambda_p \tag{3.4}$$

We have yet to define one of the main descriptors of an oscillator: its natural frequency. Since phase is mapped onto position, frequency will be mapped onto velocity. Indeed, from differentiation in time of Eq. (3.2), we have a definition of desired vehicle velocity:

$$v_i^d = \dot{\theta}_i \frac{\lambda_p}{2\pi} \tag{3.5}$$

Under this definition, it follows that the natural frequency ω_i of a vehicle is simply the frequency corresponding to the constant nominal desired velocity the vehicle would like to travel at. In our proposed strategy a key constraint is that all vehicles have the same natural frequency $\omega_i = \omega$. As such, a road segment p is characterized not only by its wavelength λ_p, but also by a nominal speed $v_{n,p}$, such that the following constraint is always satisfied[2]:

$$\omega = 2\pi \frac{v_{n,p}}{\lambda_p} \tag{3.6}$$

Now that we have a definition of phase and frequency as they relate to desired position and velocity, we consider the dynamics of this phase variable. Specifically, we impose that these dynamics be governed by the Kuramoto equation. This equation was introduced in 1975 to model the dynamics of populations of weakly coupled oscillators that exhibit self-synchronizing behaviour. Synchronization refers to oscillators with different natural frequencies influencing each other to oscillate at the same frequency and a constant phase difference. This occurs mostly in biological systems like populations of flashing fireflies. The governing equation, as proposed by Kuramoto in [16], is as follows:

$$\dot{\theta}_i(t) = \omega_i + \frac{1}{N}\sum_{j=1}^{N} K_{ij} \sin(\theta_j(t) - \theta_i(t)) \tag{3.7}$$

In this formulation, the instantaneous frequency of oscillation $\dot{\theta}_i$ is given by the oscillator's natural frequency ω_i plus the coupling term to all other oscillators based on the sine of their difference in phase multiplied by a coupling term K_{ij}.

For all-to-all symmetric coupling, that is $K_{ij} = K$, and a monotonic and unimodal distribution of natural frequencies $p(\omega)$, the behaviour and stability of the

[2] The possibility of allowing multiple nominal speeds on multi-lane road segments is not precluded by this problem formulation, since the different lanes can correspond to different wavelengths.

system is well-understood [55]. To illustrate this behaviour, it is useful to express the model in its mean-field form, by introducing the order parameter:

$$r(t)e^{\Psi(t)} = \frac{1}{N}\sum_{j=1}^{N} e^{\theta_j(t)i} \tag{3.8}$$

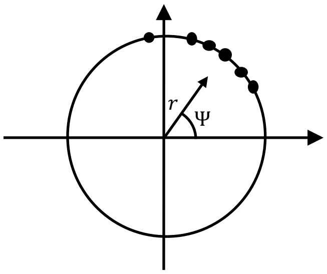


Figure 3.1: The order parameter has magnitude r (the coherence), and phase Ψ (the mean phase)

If each oscillator is thought of as a particle orbiting around the unit circle, the order parameter is the centroid of all oscillators, as shown in Fig. 3.1. The Kuramoto equation can then be rearranged in terms of r and Ψ:

$$\dot{\theta}_i(t) = \omega_i + r(t)K\sin(\Psi(t) - \theta_i(t)) \tag{3.9}$$

In this form, one can see that the i^{th} oscillator is pulled towards the mean phase Ψ with an effective coupling Kr. The coherence r takes values from 0 to 1, where 0 represents all oscillators orbiting incoherently and 1 represents all of them sharing the same phase.

When the coupling between oscillators is $K = 0$, agents orbit the unit circle in complete incoherence and the value of r fluctuates around 0. As the coupling strength is increased, incoherent behaviour persists until a critical coupling threshold

K_c is exceeded. For these larger values of K, a subset of oscillators synchronize and start recruiting more and more oscillators. Indeed, a positive relationship exists between the coherence r and the coupling strength Kr. From Eq. (3.9) we can see that the stronger the coupling, the more the oscillator is pulled towards the mean phase, and as more oscillators orbit near the mean phase, the coherence r increases. Finally, the value of r saturates at some final value below, but near 1, around which it fluctuates.

For normal distributions of natural frequencies and large enough coupling, the resulting behaviour corresponds to all oscillators orbiting with the mean frequency of the original distribution (this is called frequency entrainment) and maintaining a constant phase difference between each other (this is called phase-locking). In the particular case of all natural frequencies being the same, all vehicles phase lock to the mean phase exactly, with no constant phase difference between them, and r converges to 1 exactly. Using the frequency given in Eq. (3.6), we can write the dynamics of θ_i as:

$$\dot{\theta}_i(t) = \omega + r(t)K\sin(\Psi_i(t) - \theta_i(t)) \tag{3.10}$$

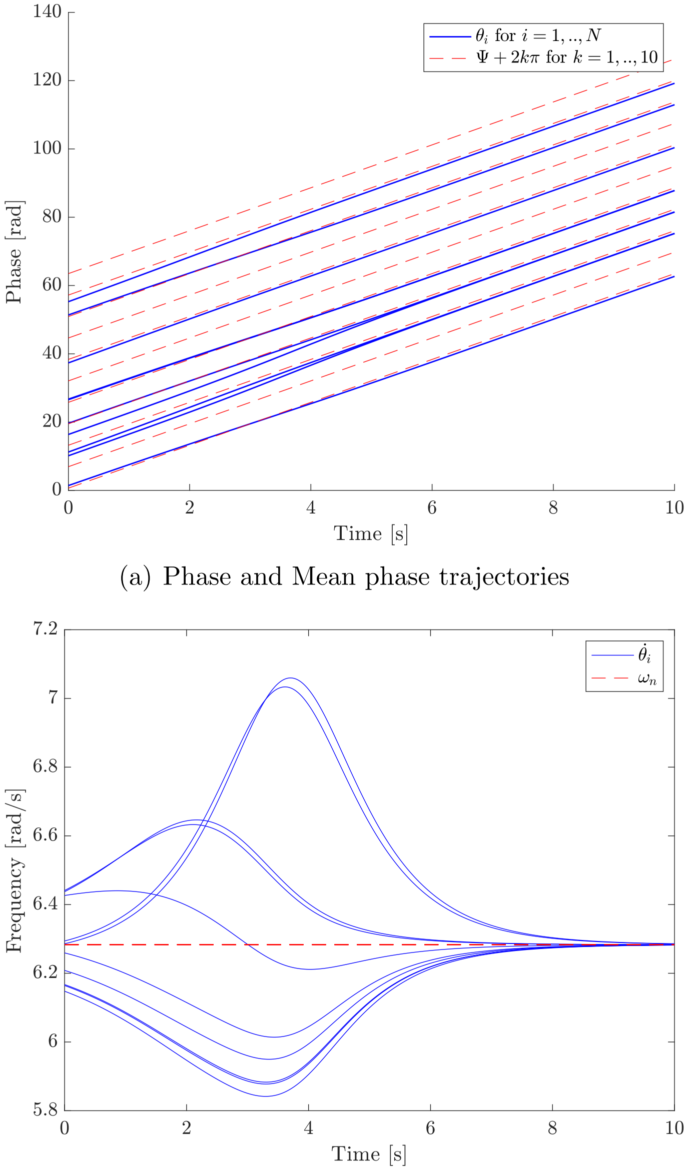
where the local mean phase Ψ_i is the closest projection to θ_i of the overall mean. That is:

$$
\begin{gathered}
\Psi_i(t) = \min_k \{\Psi(t) + 2k\pi\} \\
\text{subject to:} \\
-\pi \leq ||\Psi(t) + 2k\pi - \theta_i(t)|| \leq \pi \\
k \in \mathbb{Z}
\end{gathered}
\tag{3.11}
$$

By collapsing the distribution of natural frequencies of the oscillators into a single point (i.e $p(\omega) = 1$), we force all the phases of the system to converge to the mean phase plus some multiple of 2π, or, in other words, to its closest mean phase $\Psi_i(t)$.

For a population of oscillators with a random distribution of initial phase, the trajectories of the phase, mean phase, and frequency are shown in Fig. 3.2. Note that oscillators are basically pulled towards the closest mean phase; we can then think of the mean phase, and its projections every 2π, as beacons that the vehicles are attracted to.

Finally, we combine the behavior of a Kuramoto-driven system and the mapping between desired position and phase we have defined. This combination constitutes the coordinating layer of our algorithm. Through Kuramoto the vehicles agree on a mean phase for the entire network, and because of the definition of the mapping given by Eqs. (3.2) and (3.3), the vehicles then attempt to cross the intersection exactly when the mean phase is equal to the offset of the road. As such, the synchronizing Kuramoto layer allows vehicles to negotiate the crossing state of all intersections in the network, regardless of their distance to those intersections.



(a) Phase and Mean phase trajectories

(b) Evolution of frequencies θ_i

Figure 3.2: Evolution of phase, mean phase, and frequencies for a population of oscillators with random initial phase

3.2.2 Phase, Offset and Wavelength Constraints

Three different types of constraints need to be satisfied so that the behaviour of oscillators shown in Fig. 3.2, corresponds to solving both the intra-junction and the inter-junction problem; these are:

1. No two vehicles in the same road segment are being pulled towards the same beacon; this guarantees **spacing** between vehicles in the same road.

2. The phase offsets for intersecting roads place the vehicles in the intersection at different times; this guarantees alternate **servicing** at the intersection.

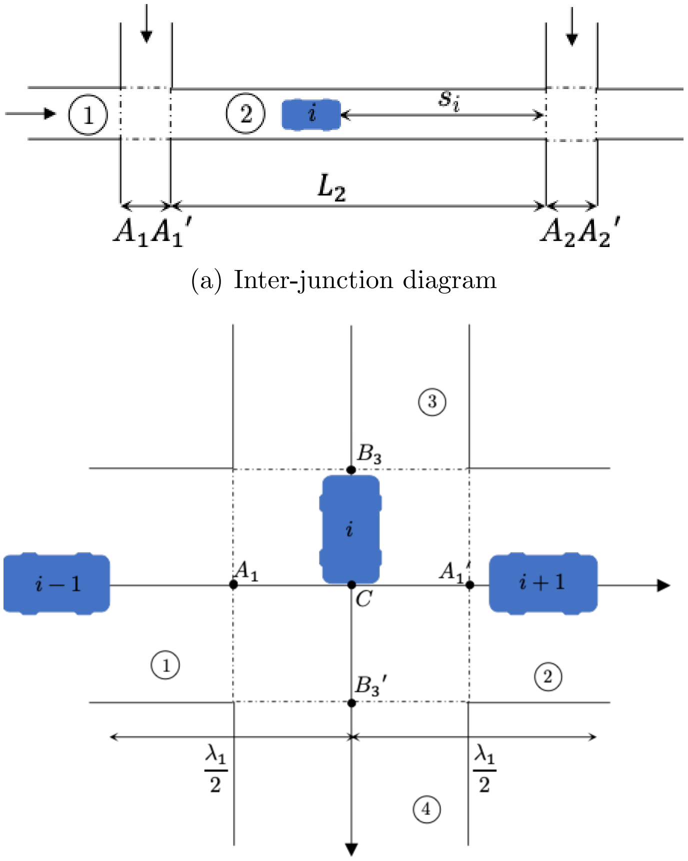
3. The phase of a vehicle as it goes from one road segment to the next segment of the same road (i.e., as it goes straight through an intersection, without turning) does not change (in the unit circle); this guarantees **continuity** of the through flow, thereby reducing energy losses due to re-synchronization.

Vehicles can meet the spacing constraint by properly correcting their phase when a conflict is detected, which mostly occurs when entering a new road segment. Recall that Kuramoto feedback pulls an oscillators towards whichever mean-phase attractor is closer to its current phase. If we define Ψ_i as the projection of the mean-phase closest to the phase θ_i of vehicle i, according to Eq. (3.11), we can write a phase resetting condition for the vehicles that guarantees the spacing constraint:

$$\theta_i = \min(\theta_i, \Psi_j - \pi - \epsilon) \quad \forall j \in \{j | s_j > s_i\} \tag{3.12}$$

where ϵ is a very small number. By saturating θ_i in this way, we make sure that $\Psi_i \neq \Psi_j$, which means that no two agents on the same road segment are pulled towards the same attractor.

To write the safe servicing and continuity constraints mathematically, we consider the variable definitions in Fig. 3.3, where we draw a representative intersection



(a) Inter-junction diagram

(b) Intra-junction diagram

Figure 3.3: Variable definition as seen within and between junctions

zone. Points A_1 and B_3 correspond to the origins of road segments 1 and 3; that is, the phase at those points is the offset of the respective road segment. Point C represents the intersecting point between the trajectories of vehicles going straight through both roads. Here, and for the rest of the chapter we consider an intersection of two one-way roads with only two conflicting traffic movements for the sake of simplicity. More practical traffic scenarios can be accounted for by partitioning the wavelength into however many flows are necessary.

The servicing constraint, which directly relates to solving the intra-junction problem, aims to maximally space out vehicles crossing the intersection from different roads. It is a constraint on the offset of each road that guarantees that each traffic flow is serviced during a different portion of the cycle. Considering the scenario drawn in Fig. 3.3 we can see that maximal spacing for vehicle i from the vehicles that cross the intersection before and after itself occurs if it reaches the intersection (i.e. point C) exactly between them. Now vehicles $i+1$ and $i-1$ are separated by a full wavelength λ_1, or by 2π radians in the phase domain (as follows from eq. (3.4)) . It follows that the distance between vehicle i and $i-1$ should be half a wavelength, or π radians in the phase domain. We can show that this is achieved if the mappings of roads 1 and 3 satisfy the following constraint, which relates the phases of point C as mapped by the mappings of each road.

$$g_1^{-1}(A_1C) = \pi + g_3^{-1}(B_3C) \tag{3.13}$$

where $g_p^{-1}(s)$ is the inverse of the mapping (3.1) for road segment p. The arguments A_1C and B_3C are the distances between each road's entrance to the intersection and the collision point. For our proposed mapping (3.2), the above equation can be rearranged as:

$$\phi_3 - \phi_1 = 2\pi(\frac{A_1C}{\lambda_1} - \frac{B_3C}{\lambda_3}) - \pi \tag{3.14}$$

Finally, the inter-junction coordination problem can be solved automatically by ensuring continuity between mappings as vehicles go from one road segment to the next. That is, we guarantee that the phases at points A_1 and B_3 are the same when mapped by roads 1 and 2, and roads 3 and 4 respectively. Recalling that points A_1 and B_3 are the origins of the intersection region, and using equations (3.2) and (3.3), this amounts to:

$$\begin{aligned} g_2^{-1}(L_2 + A_1A_1') = g_1^{-1}(0) = \phi_1 \\ g_4^{-1}(L_4 + B_3B_3') = g_3^{-1}(0) = \phi_3 \end{aligned} \tag{3.15}$$

Rearranging according to our affine mapping of equation (3.1), we can express the constraints in terms of the offsets of the roads:

$$\begin{aligned} \phi_1 - \phi_2 = \frac{2\pi}{\lambda_2}(L_2 + A_1\bar{A}_1') \pmod{2\pi} \\ \phi_3 - \phi_4 = \frac{2\pi}{\lambda_3}(L_4 + B_3\bar{B}_3') \pmod{2\pi} \end{aligned} \tag{3.16}$$

It is worth noting that Eq. (3.16) can only partially guarantee continuous flow as vehicles travel along a corridor of intersections. For one, the constraint cannot be imposed to turning flows, since the servicing constraint ensures the destination road segment of a turning vehicle will be π radians out of phase with respect to its road of origin. Another scenario where flow is disrupted occurs when another vehicle turns into the destination section of the vehicle going straight. In this situation, because of the spacing constraint, the latter vehicle will be forced to slow down to catch the upstream wave. Finally, while the wavelengths can be thought of as adjustable variables in constraint (3.16), wavelengths are also constrained by their relationship with frequency and velocity through Eq. (3.6). Specifically, a change in wavelength from one section to the next would force a change in desired speed through Eq. (3.6) in order to maintain a constant natural frequency, creating an undesirable acceleration or deceleration event. For the rest of this work, we therefore assume that wavelengths and desired speeds are the same across all roads in the network, and we drop the road identifying index p for λ and v^d.

In guaranteeing spacing, safety and continuity to solve the coordination problem at both scales, we have introduced two different types of constraints. The spacing constraint (3.12) is a constraint on the actual phase of the vehicles; it forces vehicles to push their desired phase back, and with it the time at which it will cross the intersection. This constraint needs to be checked for and implemented continuously, although it will mostly become active when vehicles change road segments. The servicing and continuity constraints, on the other hand, are constraints on the constant design variables of the network, namely the offsets and wavelengths of the roads, and they are chosen before any vehicles enter the network. Along with the desired speed v^d, these design variables determine the maximum throughput of the network as we will discuss in subsequent sections.

3.2.3 Optimal mean-phase tracking

So far we have discussed the dynamics of a vehicle's desired phase, which is then mapped to a desired position. In previous work [50], we propose a linear feedforward/feedback tracker that uses this signal as reference. Further insight into the behaviour of the system of coupled oscillators allows us to propose here a more sophisticated tracking approach, namely, a model predictive optimal controller that minimizes the jerk of vehicles using predictions of both the arrival time imposed by the phase dynamics and the behaviour of other vehicles.

We can show that the synchronizing layer described above determines the time $\tau_i(t)$ at which the vehicle i should ideally arrive at the intersection. Indeed, the computation of $\tau_i(t)$ follows from the properties of Eq. (3.9), where the mean-phase Ψ oscillates with a constant frequency ω [55].

$$\dot{\Psi}(t) = \omega \tag{3.17}$$

As described in the previous section, vehicle i should reach the intersection when its phase is already tracking its mean phase beacon, which is in turn equal to the offset of the road:

$$\theta_i(\tau_i) = \Psi_i(\tau_i) = \phi_p \tag{3.18}$$

It follows from the previous two equations that for vehicle i at time t the expected time of arrival at the intersection is given by:

$$\tau_i(t) = \frac{\phi_p - \Psi_i(t)}{\omega} \tag{3.19}$$

Since the vehicle enters the intersection at time τ_i, in synchrony with its mean phase beacon, its desired position, velocity and acceleration are also known:

$$
\begin{aligned}
s_i(t+\tau_i) &= 0 \\
v_i(t+\tau_i) &= v^d \\
a_i(t+\tau_i) &= 0
\end{aligned}
\tag{3.20}
$$

Assuming vehicles can control their jerk, or their change in acceleration, through accurate lower level powertrain and vehicle dynamics controllers, we model these vehicles as third order dynamical systems. Note that we choose a third order system here, instead of the second order system traditionally used to model vehicles, because it will yield smoother acceleration profiles. With the third order model, the state variables for each vehicle are then: (i) its distance to the intersection, along the path of the road; (ii) its velocity; and (iii) its acceleration. The input is the jerk of the vehicle:

$$
\begin{aligned}
\dot{s}_i(t) &= v_i(t) \\
\dot{v}_i(t) &= a_i(t) \\
\dot{a}_i(t) &= u_i(t)
\end{aligned}
\tag{3.21}
$$

The control input $u_i(t)$ that places the vehicle at the intersection at the right time, can be the solution of an optimization problem that minimizes mean square jerk:

$$\min \int_{t}^{t+\tau_i(t)} \frac{1}{2} u_i(\tau)^2 d\tau \tag{3.22}$$

Subject to:

State dynamics (3.21)

Terminal time conditions (3.20)

$$\begin{gathered} s_j(t) - s_i(t) - S \leq 0 \\ a_{i,\min} \leq a_i(t) \leq a_{i,\max} \\ v_{i,\min} \leq v_i(t) \leq v_{i,\max} \end{gathered} \tag{3.23}$$

The additional inequality constraints guarantee that the vehicle stays a safe distance S from its leading vehicle j, and that the acceleration and velocity are bounded.

The solution to the problem without the inequality constraints (3.23) can be determined analytically by performing a Hamiltonian analysis. This approach is similar to the work of Malikoupoulos et al. in [30, 58], where the solution to a second order dynamical system, where the input is acceleration rather than jerk, is presented. In our case, the optimal trajectories for the input and the states, denoted with an asterisk, are given by:

$$
\begin{aligned}
u^*(t) &= -\frac{1}{2}c_1t^2 + c_2t - c_3 \\
a^*(t) &= -\frac{1}{6}c_1t^3 + \frac{1}{2}c_2t^2 - c_3t + c_4 \\
v^*(t) &= -\frac{1}{24}c_1t^4 + \frac{1}{6}c_2t^3 - \frac{1}{2}c_3t^2 + c_4t + c_5 \\
s^*(t) &= -\frac{1}{120}c_1t^5 + \frac{1}{24}c_2t^4 - \frac{1}{6}c_3t^3 + \frac{1}{2}c_4t^2 + c_5t + c_6
\end{aligned}
\tag{3.24}
$$

The constants $c_{1,\ldots,6}$ in the above equations are integration constants, and they can be solved for by imposing initial and final time conditions. The initial conditions are given by the current state of the vehicle at time t, and the final conditions are given in equation (3.20). The resulting system of equations is linear, and it is solved by inverting a 6-by-6 matrix and multiplying it by the concatenated vector of initial and final conditions.

The above is the solution to the unconstrained problem; the solution to the constrained problem can be determined numerically by discretizing and using a quadratic programming solver. This type of optimization is convex, however it requires full knowledge of the planned trajectory of the lead vehicle. In this chapter, our proposed solution method consists of computing the analytical solution to the unconstrained problem, and checking for constraint activity. If no constraint is infringed upon by the analytic unconstrained solution, we execute the computed input trajectory. Otherwise, we use this candidate solution as the initial guess to the quadratic programming solver and implement the constrained solution instead. In the next chapters we explore how the analytic solution can be modified in order to satisfy safety and feasibility constraints without the need to know or compute the full trajectory of the leading vehicles.

3.2.4 Summary

To summarize the workings of our algorithm, let us recount the actions vehicle i takes at any given time t, after it receives the phase and mapping information from the rest of network:

1. If the vehicle has just entered a new road segment, it selects its initial phase $\theta_i(t)$ to match its current position according to the mapping of the road.

2. It computes the order parameter of the system of oscillators (i.e the mean-phase Ψ and coherence r of the network), as well as the projection Ψ_i of Ψ closest to its own phase.

3. If both its phase and that of its leading vehicle j are most proximal to the same mean-phase beacon ($\Psi_i = \Psi_j$), the vehicle pushes its phase backwards by $\pi + \epsilon$ radians from the beacon tracked by its leader ($\theta_i = \Psi_j - \pi - \epsilon$).

4. It computes the time it should arrive at the intersection $\tau_i(t)$ given the current mean-phase.

5. It determines the optimal trajectory of its state and input that minimizes jerk, according to the analytical solution to the unconstrained optimization problem.

6. If the solution violates constraints, it solves the constrained optimization problem numerically.

7. It updates the value of its phase through the Kuramoto equation.

8. It implements the first input command according to the generated input trajectory.

Fig. 3.4 summarizes this process in a block diagram. The result of following this protocol is that vehicles cross the intersection at different times, and that acceleration maneuvers as they go from one intersection to next are not very aggressive.

3.3 Designing for Traffic Flow and Safety

Before looking at the performance of our strategy in simulation, we can discuss some of its anticipated implications in terms of traffic flow and density. By virtue of Kuramoto synchronization, all vehicles oscillate at the same frequency once coherence is achieved. In fact, in our current formulation, this frequency corresponds to the natural frequency we choose for the network:

$$\omega = 2\pi \frac{v_{n,p}}{\lambda_p} \tag{3.25}$$

The flow of vehicles in each road is directly related to this frequency given that vehicular flow is the product of velocity and density. Maximum density is nothing more than the inverse of the wavelength, because in each road there can only be one vehicle per wave, and vehicles are spaced by one wavelength (or more). The

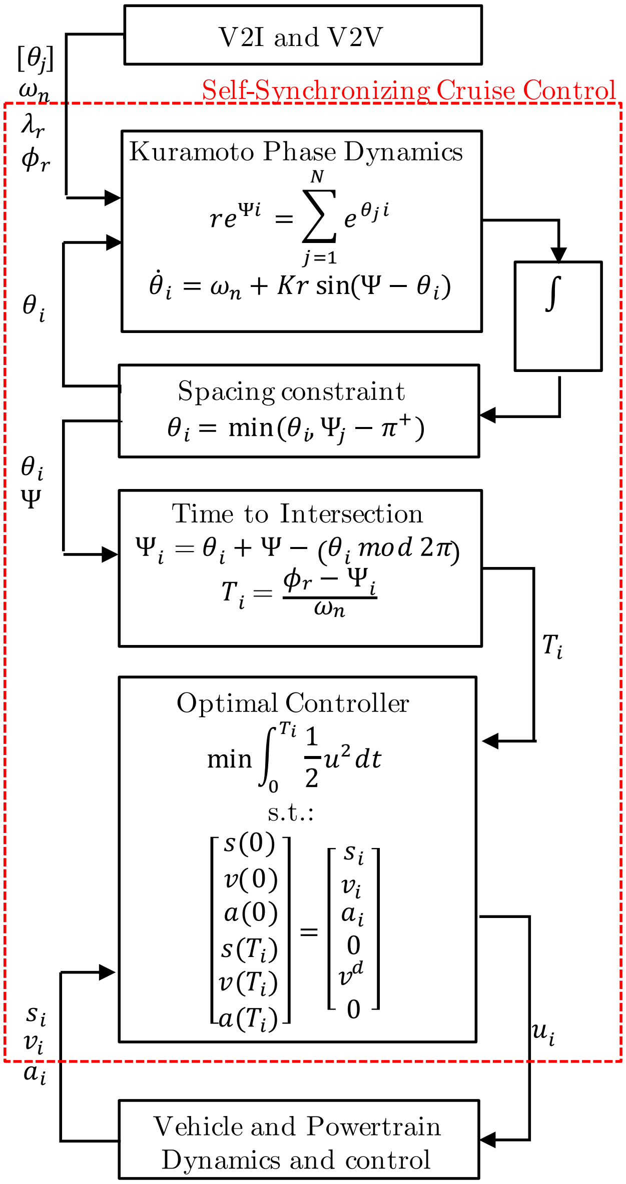

V2I and V2V
Self-Synchronizing Cruise Control
$[\theta_j]$
ω_n
λ_r
ϕ_r
Kuramoto Phase Dynamics
$re^{\Psi i} = \sum_{j=1}^{N} e^{\theta_j i}$
$\dot{\theta}_i = \omega_n + Kr\sin(\Psi - \theta_i)$
$\int$
θ_i
Spacing constraint
$\theta_i = \min(\theta_i, \Psi_j - \pi^+)$
θ_i
Ψ
Time to Intersection
$\Psi_i = \theta_i + \Psi - (\theta_i \bmod 2\pi)$
$T_i = \frac{\phi_r - \Psi_i}{\omega_n}$
T_i
Optimal Controller
$\min \int_0^{T_i} \frac{1}{2} u^2 dt$
s.t.:
$\begin{bmatrix} s(0) \\ v(0) \\ a(0) \\ s(T_i) \\ v(T_i) \\ a(T_i) \end{bmatrix} = \begin{bmatrix} s_i \\ v_i \\ a_i \\ 0 \\ v^d \\ 0 \end{bmatrix}$
s_i
v_i
a_i
u_i
Vehicle and Powertrain Dynamics and control

Figure 3.4: Kuramoto Coordinator Control Architecture

maximum possible flow is then, in its traditional units of vehicles per hour:

$$q = \frac{v_{n,p}}{\lambda_p}(3600) = \frac{\omega_n}{2\pi}(3600) \tag{3.26}$$

We can then expect that for input flows below the selected natural frequency, the algorithm will be able to meet the traffic demand. For higher input flows, a queue will start to form at the entrances of the network as vehicles wait to track non-occupied wave crests.

Velocity and wavelength should be chosen to produce a natural frequency higher than the demand of the road. However, this is not the only constraint on these two variables, since the spacing of vehicles as they cross the intersection also depends on these variables. In fact, from analysing Fig. 3.3, we can determine that the gap in seconds between a vehicle at the collision point of the intersecting paths and the vehicle that just crossed is given by:

$$G = \frac{1}{v_{n,p}}(\frac{\lambda_p}{2} - S) \tag{3.27}$$

Where S is a safety distance that needs to be larger than the occupied portion of the wave, that is, the length plus the width of the vehicles.

Having defined the relationship between our design variables $v_{n,p}$ and λ_p, we can look at the inherent trade-offs between increasing the maximum throughput of the network and maintaining enough spacing between vehicles at the intersection. Fig. 3.5 shows this trade-off in our design space. We have drawn lines of constant throughput and lines of constant safety gap. We can see that to increase the safety gap, one might decide to choose larger wavelengths; however, since this will reduce the density of the roads, throughput will be affected. Alternatively, if one wishes to increase throughput, the simplest way to "cross" dashed-blue lines is to increase speed, but this comes at the cost of reducing the safety gap. Another inherent trade-off that does not show up in Fig. 3.5, as it is more difficult to compute analytically,

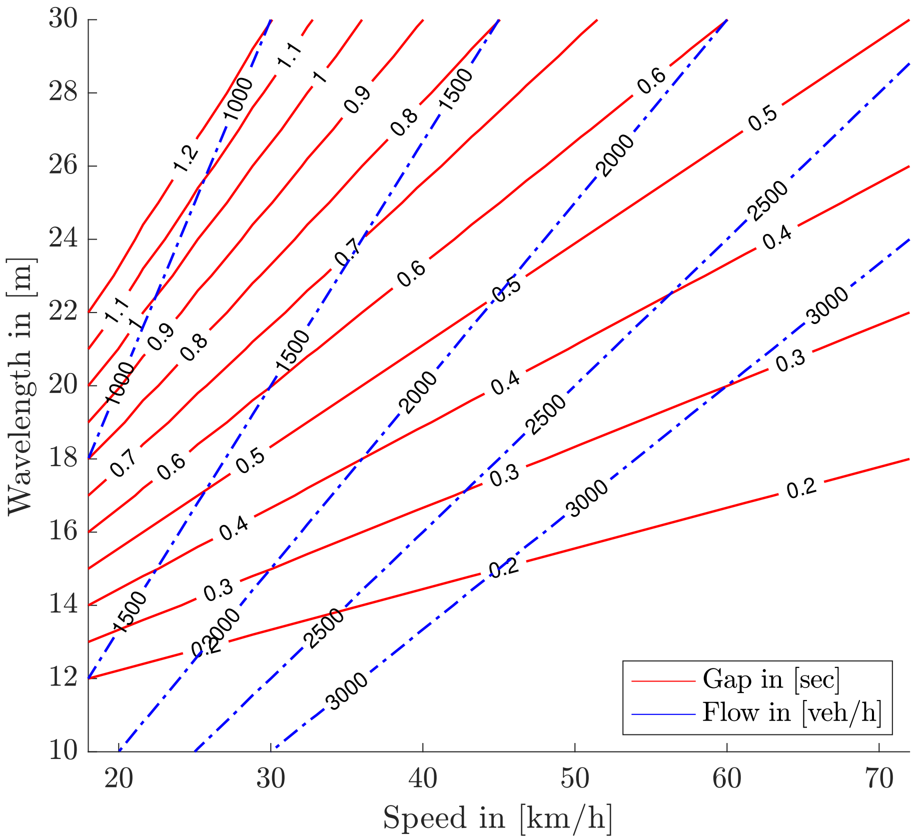


Figure 3.5: Relationship between speed, wavelength, flow and safety

is the energy/fuel cost associated with having longer wavelengths. If a vehicle enters a road completely out of phase, the acceleration/deceleration maneuver it will need to perform is larger in roads with larger wavelengths. This translates to a higher kinetic energy change, and with it, some potential waste of fuel.

3.4 Simulation Results

In this section, we study the performance and characteristics of our proposed strategy in simulation. We consider a network of one-way roads consisting of 9 intersections and 24 road segments where vehicles can either go straight or turn right; a snapshot of this network is shown in Fig. 3.6. The road segments are approximately

90 meters long, and the straight segment of the intersections is approximately 10 meters long. The entry roads to the network are assumed to be longer, at 200 meters. For this network, where the origins of the mappings between connected road segments are 100 meters apart, a 20 meter wavelength would satisfy the continuity and servicing constraints of Eqs. (3.16) and (3.14) if we choose offsets of 0 and π for horizontal and vertical roads respectively.

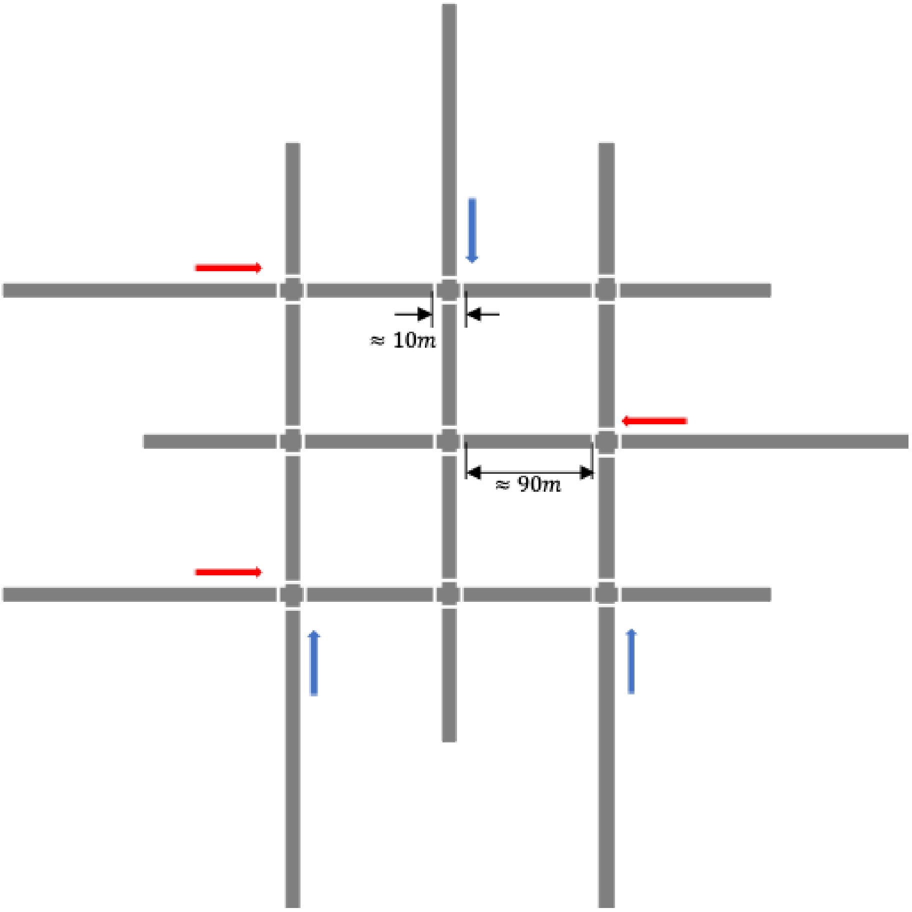


Figure 3.6: Network of 9 intersections used in simulation.

3.4.1 State Trajectories

Fig. 3.7 shows the distance to the intersection as a function of time for a group of vehicles approaching the intersection at the center of the network. In this figure, as in subsequent ones, the color of the curve indicates whether the vehicle is travelling down a horizontal (dashed red) or a vertical (solid blue) road segment, and, for clarity, we flip the sign of the distance along the horizontal directions. We can see that red and blue lines cross the 0 line at different points, meaning that vehicles enter the intersection at different times. Moreover, the plot illustrates how vehicles space out evenly along the same road.

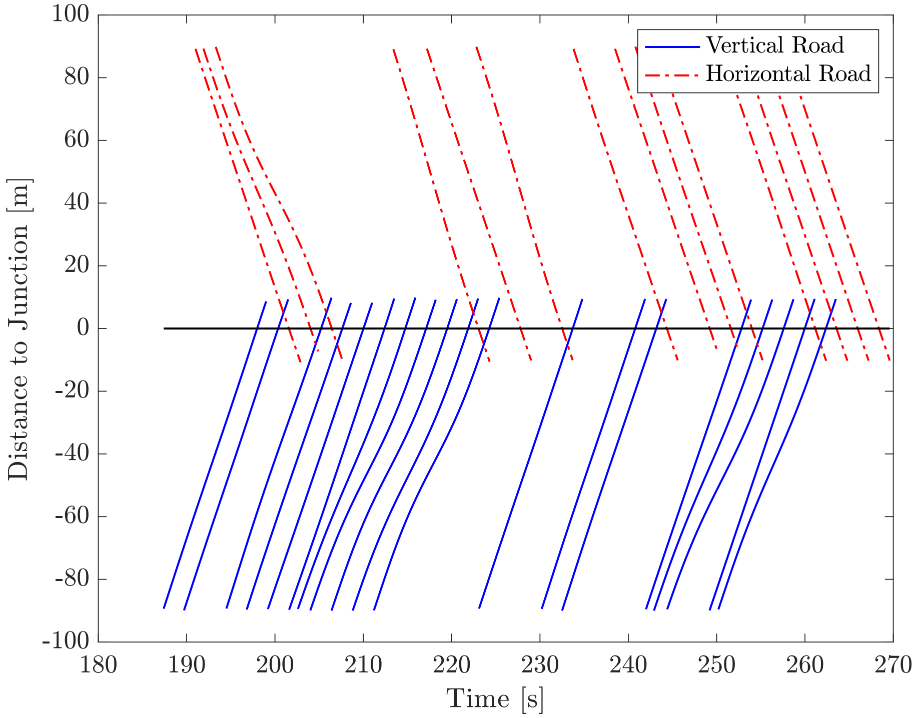


Figure 3.7: Example position trajectories for a group of vehicles approaching the same intersection along the horizontal (red dotted-solid line) and vertical (blue solid line) roads.

We can also look at the position, velocity and acceleration of a single vehicle as it travels through the network, which we show in Fig. 3.8. Here, we have also plotted in solid blue the vertical segments, and in dashed red the horizontal ones. We can see that as the vehicle goes straight through the intersections its velocity profile stays relatively flat, as promoted by the continuity constraint we impose on the mapping and the fact the consensus occurs at a network level. When the vehicle turns in the third intersection it needs to adjust its speed to match the offset of the new road it travels on. The same thing happens as it turns right again in the next intersection.

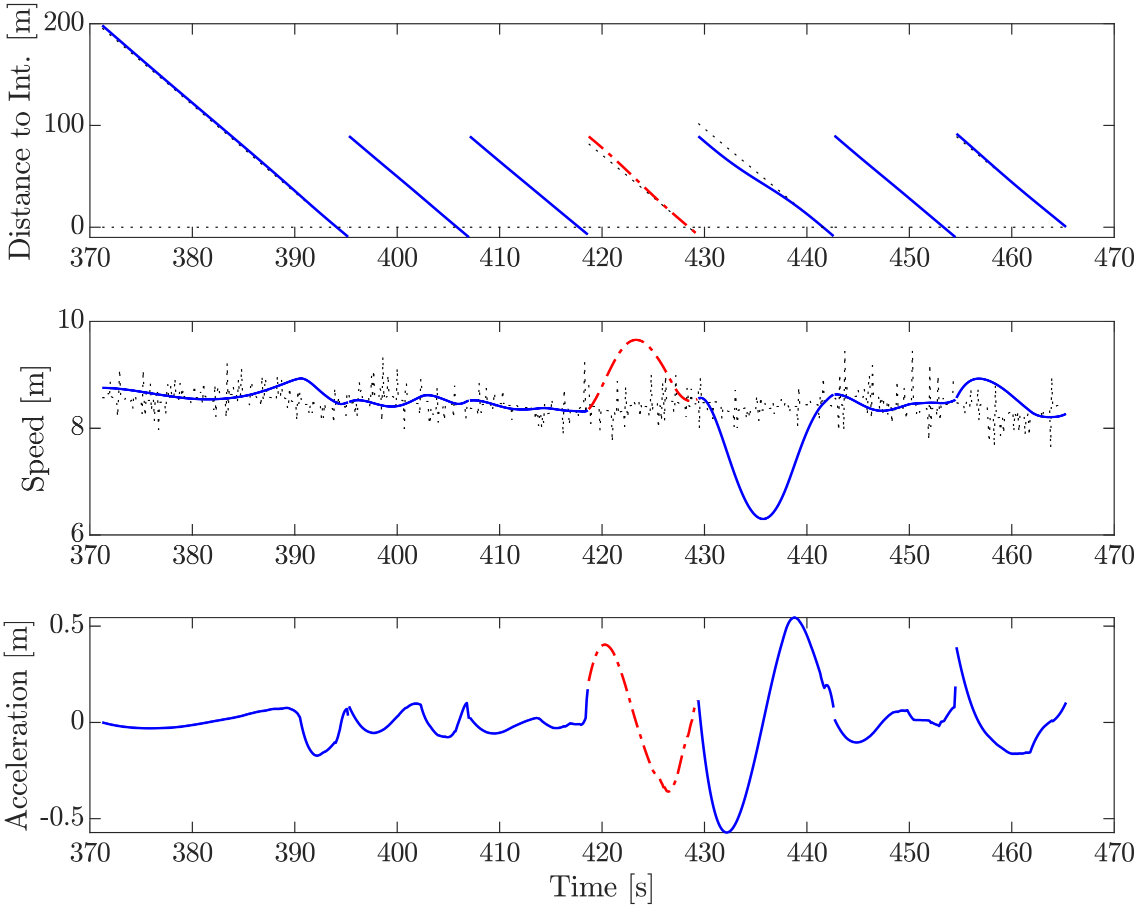


Figure 3.8: Example position, velocity and acceleration trajectories for a single vehicle travelling through the network in horizontal (red solid-dotted) or vertical roads (solid blue), along with the reference mean phase and frequency (dotted black).

3.4.2 Fuel Consumption and Delay Time Results

We can evaluate the fuel consumption and delay time of vehicles using our strategy compared to simulated human drivers controlled by traffic lights. The baseline drivers are governed by a modified Gipps car following model [52] as implemented in Aimsun, an established traffic simulator. We set the input flow of all entry roads at 750 vehicles per hour, with a turn percentage of 20%. The arrival process of vehicles into the network is the main source of stochasticity in our simulation, and it is modeled as a Poisson arrival process, as is traditionally done in traffic simulation [**?**]. We choose a traffic light cycle of 60 seconds, with 25 seconds of green time for each flow and 10 seconds of clearing time. Furthermore we offset the green time of the lights in pursuit of the ”green wave” effect, which occurs when vehicles catch several green windows in a row as they travel down an arterial corridor. We run the baseline simulation for 10 minutes of simulated time, and we replicate the scenario with the same vehicle injection times and paths, but using our Kuramoto strategy instead. Fig. 3.9 shows the baseline position trajectories corresponding to the same vehicles shown in Fig. 3.7 in the last sub-section.

The described simulation consists of about 750 vehicles, but for our comparisons we consider only the 100^{th} through 600^{th} vehicles. In this way, we allow for the network to build some capacity, and we do not consider vehicles who do not finish their path before the simulation is stopped.

We are interested in looking at two metrics relevant for traffic performance evaluation: fuel consumption and delay time. The delay time is simply the difference

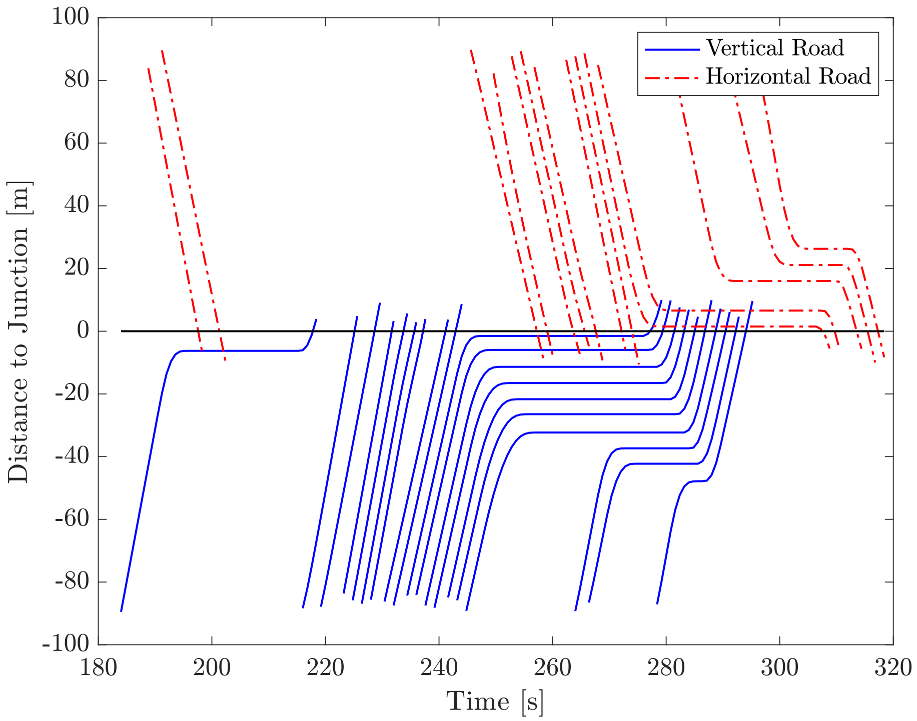


Figure 3.9: Example baseline position trajectories for vehicles approaching an intersection controlled by a traffic light.

between a vehicle's travel time and its corresponding minimum travel time had it cruised at the desired speed of the road, normalized by the total distance traveled. To calculate fuel consumption, we use a fuel map for a 1.2 liter gasoline engine. The map translates every engine torque and speed pair to a fuel rate $\dot{m}_f$. To use it, we first calculate the wheel power required to meet the acceleration and velocity trajectories imposed by the driver. We then estimate the corresponding engine power by assuming an efficiency ratio for the transmission. Finally, we say the vehicle uses the minimum fuel rate associated with this engine power demand, which assumes the transmission can operate at the required engine torque and speed combination. Fig. 3.10 shows the optimal fuel rate vs. engine power line we get.

Fig. 3.11 shows the delay time and fuel consumed by each of the vehicles for both the baseline and proposed scenario. When we compare the average of both point clouds, we find that our proposed strategy leads to a 48% and 57% reduction in fuel consumption and delay time respectively for this particular scenario. Furthermore, we note a significant reduction in the spread of the point cloud, meaning that there is less variability in the anticipated behaviour of the vehicles. Indeed, in the baseline, a vehicle that encounters a desirable green wave of traffic light sequences can traverse the network quickly without stop-and-go behaviour, whereas vehicles that are less lucky are forced to stop at several intersections in sequence.

If we compute the work done by negative propulsive forces (i.e. braking), drag forces, and rolling resistance forces in our model for longitudinal vehicle dynamics, we can see where energy losses are incurred. The savings in fuel consumption can then indeed be attributed to a reduction in energy losses due to braking. In other words, our strategy improves performance by reducing stop-and-go behaviour, as expected. Fig. 3.12 shows the result of this energy balance.

3.5 Conclusion

In this chapter we present a control strategy for connected and autonomous vehicles that solves the intersection coordination problem in both of its scales. That is, our strategy synchronizes vehicles crossing the same intersection, and it smooths the flow from one intersection to the next. This is achieved by defining a mapping between a vehicle's position and its corresponding phase in a virtual system of oscillating agents coupled by the Kuramoto equation. The mapping, with its safety constraints within the intersection and continuity constraints between intersections, guarantees the desired behaviour of the reference position. This reference is then tracked through an optimal control problem that is first solved analytically, and then numerically if constraints are violated. The resulting strategy saves both fuel and travel time, and reduces the variability in these metrics seen across the fleet.

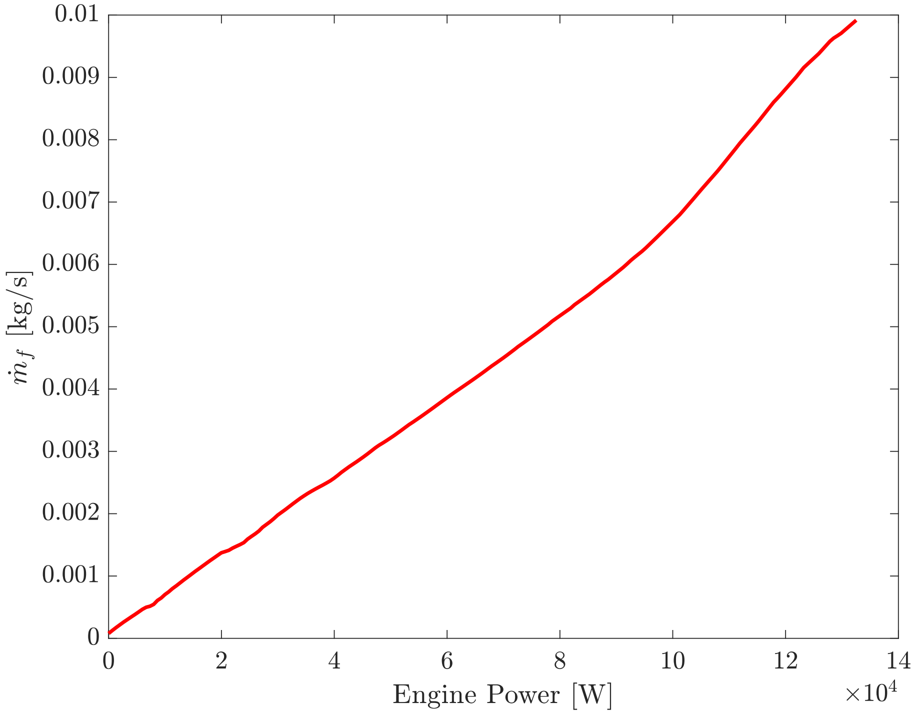


Figure 3.10: Optimal fuel rate vs. engine power for a 1.2 liter gasoline engine

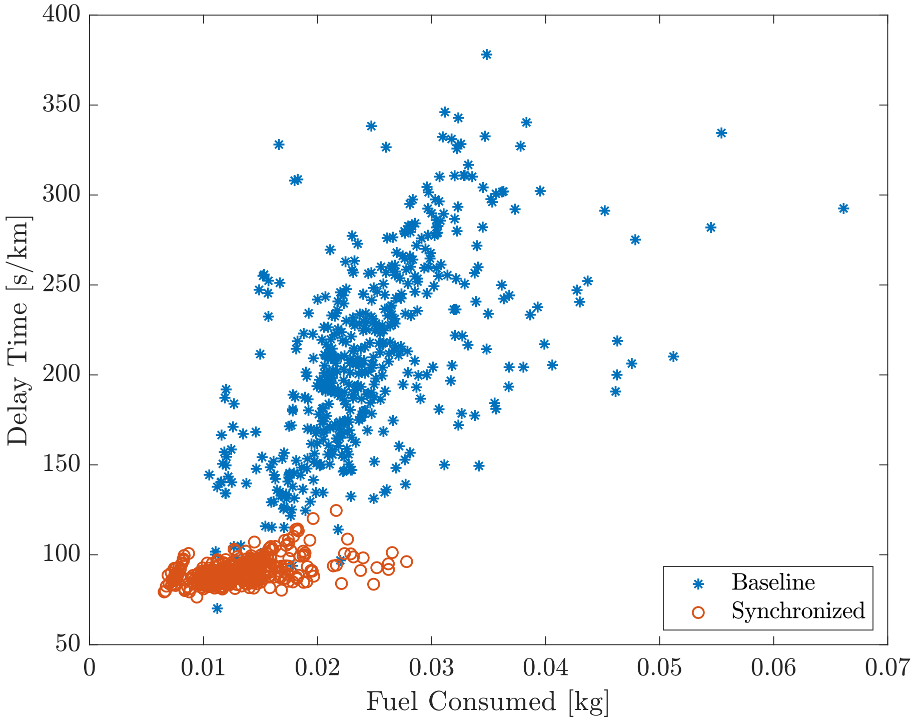


Figure 3.11: Delay time and fuel consumed for each vehicle in the simulation of the baseline (green) and proposed strategy (blue)

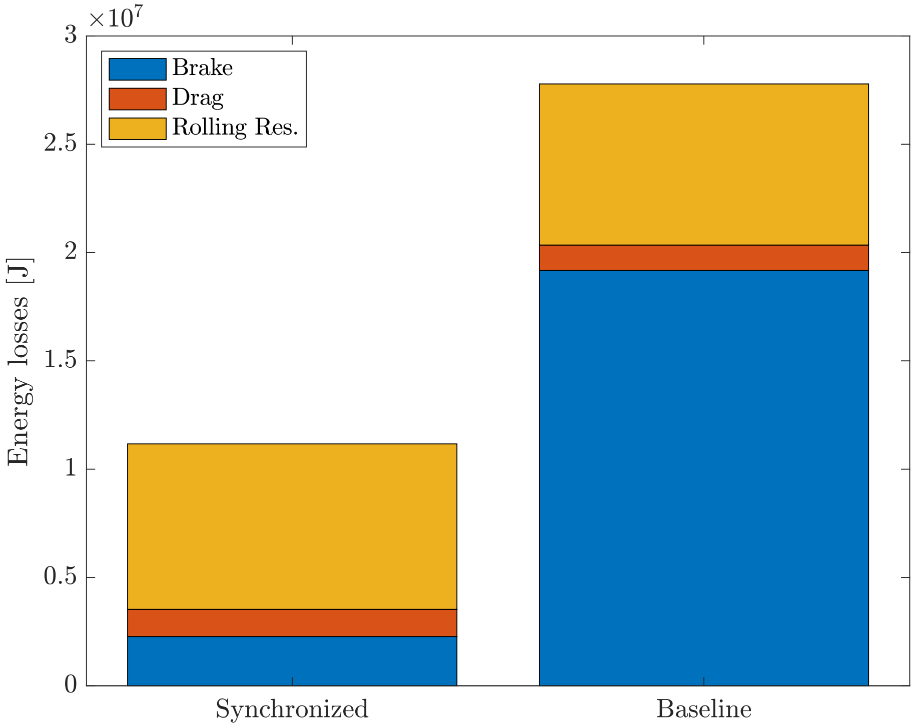


Figure 3.12: Energy losses by type (brake, drag, and rolling resistance) for the synchronization strategy and the baseline strategy

Chapter 4: A Gradient-Based Approach for Coordinating Smart Vehicles and Traffic Lights at Intersections

4.1 Introduction

[1] The idea behind the strategy presented in the previous chapter can be generalized beyond the Kuramoto model. Indeed the control architecture can be described as follows: a coordinating layer governs the planned timing of the vehicles, and a tracking layer attempts to achieve the current planned timing. Given this generalization, in this chapter we present a broader method for describing the interaction between agents using Edge Tension Functions (ETF) within a gradient-based control law. For the right ETF, the Kuramoto model can be thought of as a gradient-based multi-agent controller, and like the Kuramoto model, the complexity of gradient-based coordination grows linearly with the number of neighbors that agents communicate with [60]. In terms of the coordination variable, here, we move from describing vehicles by their phase, which was then related to their planned arrival time, to directly describing them by their planned arrival time. Furthermore, the approach presented in this chapter also includes traffic lights as controlled agents,

[1] The work presented in this chapter is adapted from [59], accepted in *IEEE Control Systems Letters, (2021)*

described by their planned switching times.

Specifically, we consider both the traffic light and the approaching vehicles as smart connected agents and we define a suitable edge tension function in terms of their planned timing for each connection between agents. The agents then run a decentralized gradient descent control policy that drives them towards a desirable sequence of intersection crossing times. As we have seen in previous chapters, the problem of coordinating CAVs at intersections has been widely explored, with different algorithms being proposed.

From the perspective of optimal control and multi-agent system control, most of the work considers the problem of coordinating connected and autonomous vehicles without the use of traffic light signals. For extensive literature reviews covering the different proposed approaches, we refer back to Chapter 1. The most common architecture for the proposed control strategies distributes the problem into a two-layer hierarchy, where the upper layer determines vehicle crossing times, and the lower layer determines the optimal speed trajectory vehicles need to follow to arrive at the time determined by the upper layer. We can distinguish the different strategies under this category by the methods they use to coordinate arrival times and the objectives they optimize at the vehicular level. In [53], arrival times are negotiated in a decentralized manner using the Kuramoto nonlinear consensus equation. In [28], a combination of optimization and heuristic rules such as maintaining first-in-first-out crossing sequences and grouping vehicles into bubbles are used to determine their crossing times.

The approaches considered above do not make use of traffic light signals. When it comes to coordination of both traffic lights and CAVs, the problem has been mostly explored through optimization methods, as we summarize in Table 1.1 in Chapter 1. References [45,47,49] use a microscopic model of traffic, and tackle the problem using centralized optimization for individual intersections. The trajectory planning problem is simplified through heuristic approximations and included as a subroutine of an arrival and switching time optimization problem. In [49], Tajalli et al. use a cell-transmission model of traffic to model a network of intersections and formulate an optimal coordination problem that minimizes speed variations and maximizes throughput. Through insightful relaxations, the problem is solved online in a decentralized manner. While these optimization approaches can yield significant performance improvements, coordinating both traffic lights and vehicles can be computationally prohibitive, especially as we consider more agents and larger networks.

In this chapter we propose the use of decentralized gradient-based multi-agent coordination of both CAVs and traffic lights. As such, the main contributions of this work are:

- A hierarchical control architecture that allows vehicles and traffic lights to negotiate their timing in a decentralized and computationally efficient manner, through the use of edge tension functions (ETF), while considering feasibility and safety constraints.
- A candidate ETF that drives the agents to desirable behaviours.
- Simulation studies that show the ability of the strategy to adapt to varying

traffic demands, different levels of CAV penetration, and different intersection geometries.

4.2 Proposed Strategy

In this section we present our proposed solution to the problem of controlling CAVs and smart traffic lights at single intersections. The approach consists of representing both the vehicles and traffic light by their planned time to arrival or time to switching. The agents then modify their planned timing through a decentralized gradient-based control law, with suitable bounds on the possible arrival and switching times. Then, vehicle agents analytically determine an acceleration-minimizing input trajectory, based on their current planned time to arrival. This trajectory is implemented if it is deemed safe, otherwise the vehicle saturates its input based on safe inter-vehicle spacing and traffic light constraints.

4.2.1 Problem Formulation

We consider an intersection controlled by a traffic light, with R lanes feeding into it. We assume that there exists a control zone that extends a distance L from the stop line along the length of each lane.

4.2.1.1 Traffic Light Model

We assume that the actuated traffic light follows a predetermined sequence of P phases, where each phase is built by grouping a subset of compatible movements at the intersection. The traffic light switches from flashing green for a given phase group to the next in the sequence at specific time instants. At all time instants k, we consider the set $\mathcal{L}(k)$ of all of the traffic light's desired times till switching (DTTS) over the upcoming considered time horizon T_H. Each DTTS $i \in \mathcal{L}(k)$ is given by $\tau_i(k)$. When $\tau_i(k)$ reaches 0 the traffic light switches from one phase to the next. More specifically, the light switches to yellow for the current phase, and after a set amount of yellow time τ_{yellow}, the light switches to enable the next phase group in the sequence. The evolution of $\tau_i(k)$ is given by:

$$\tau_i(k+1) = \mathcal{S}_{T_i(k+1)}(\tau_i(k) - h + u_i(k)) \tag{4.1}$$

where h is the simulation time step, $u_i(k)$ is a control input the traffic light can implement to modify its planned timings, and $\mathcal{S}_{T_i}(\tau)$ is a saturation function onto the domain $T_i(k) = [\tau_{i,\min}(k), \tau_{i,\max}(k)]$:

$$\mathcal{S}_{T_i(k)}(\tau) = \begin{cases} \tau_{i,\min}(k) & \tau < \tau_{i,\min}(k) \\ \tau & \tau \in [\tau_{i,\min}(k), \tau_{i,\max}(k)] \\ \tau_{i,\max}(k) & \tau > \tau_{i,\max}(k) \end{cases} \tag{4.2}$$

The bounds on each DTTS are set so that they do not cross each other, and the sequence of phases is preserved. That is, $\tau_{i,\min}(k) = \tau_{i-1}(k)$ and $\tau_{i,\max}(k) = \tau_{i+1}(k)$, where the $i-1$ and $i+1$ indices refer to the desired times till switching that are right before and after the i^{th} DTTS.

When the DTTS reaches 0, it is removed from $\mathcal{L}$. Moreover, a new DTTS is spawned at $\tau_i(k) = T_H(k)$ when the latest planned switch reaches a minimum green interval $G_{\min,p}$, associated with the intended phase p at T_H.

4.2.1.2 Vehicle Model

At any time instant k, we consider the set $\mathcal{V}(k)$ of vehicles approaching the intersection. Each agent $i \in \mathcal{V}$ is modeled as a discrete second-order dynamic system with synchronous time step h, so that its longitudinal dynamics are given by:

$$\begin{aligned} s_i(k+1) &= s_i(k) + hv_i(k) + \frac{1}{2}h^2 a_i(k) \\ v_i(k+1) &= v_i(k) + ha_i(k) \end{aligned} \tag{4.3}$$

where s_i is the vehicle's station distance along its lane, v_i is its longitudinal speed, and a_i its longitudinal acceleration. Note that the double integrator model used here can be obtained from a more traditional longitudinal point-mass vehicle dynamics model through feedback linearization. To represent constraints imposed by the road and the vehicle's powertrain, we assume bounds on velocity and acceleration, which for simplicity are set to be the same for all vehicles:

$$\begin{aligned} v_{\min} \leq v_i(k) \leq v_{max} \\ a_{\min} \leq a_i(k) \leq a_{max} \end{aligned} \tag{4.4}$$

Moreover, we can define a bound on acceleration based on a safe following headway with the vehicle in front (or the stop line of the intersection, given a red light). Indeed, to avoid rear-end collisions and stop at red lights a vehicle's velocity at the next time step is bounded above by some safe velocity $v_{i,b}(k+1)$, which can be a function of the current velocity of the vehicle, and the velocity and position of its leader (in the case of a red light, the leader can be treated as a virtual vehicle stopped at the intersection stop line). As such the acceleration must satisfy:

$$a_i(k) \leq \frac{1}{h}(v_{i,b}(k+1) - v_i(k)) \tag{4.5}$$

In traditional car following models the vehicles follow a free flow acceleration control law:

$$a_{i,ff}(k) = \frac{1}{h}(v_{i,a}(k+1) - v_i(k)) \tag{4.6}$$

From equations (4.3) through (4.6) we can simulate human driven vehicles. Indeed, we can use the Gipps car following model to obtain the expressions for $v_{i,b}$ and $v_{i,a}$.

4.2.1.3 Multi-Agent Network Model

We assume that both the vehicles and the traffic lights are capable of communicating state information with each other, thereby forming a network of interconnected agents. This network can be represented by an undirected graph where each node represents an agent and each edge represents the ability of neighboring agents to exchange state information. An example of the topology of the network is shown in Fig. 4.1. By assumption, the vehicles traveling down the same lane are connected, and each vehicle is connected to the traffic light.

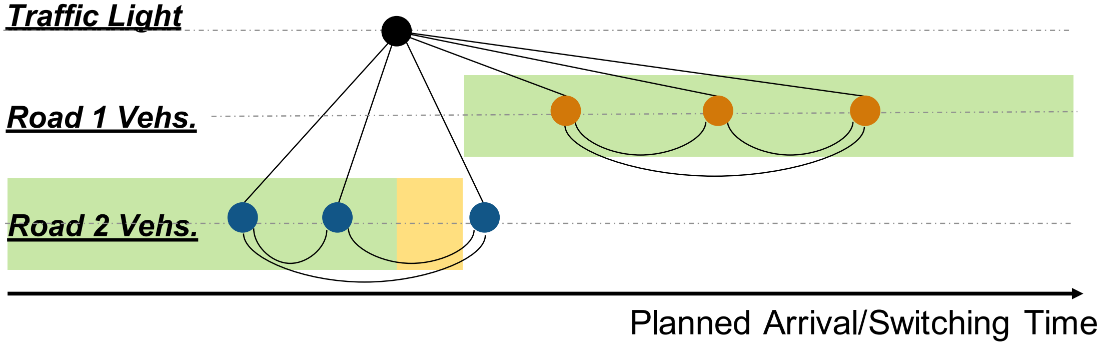


Figure 4.1: Illustrative example of graph connectivity

We represent both types of agents, vehicles and traffic lights, by a virtual state variable that governs when vehicles arrive at the intersection and when the light changes phases. For the traffic light we use the previously defined desired time till switching (DTTS). The traffic light can have multiple switches planned in the considered time horizon. To account for this, we think of each DTTS as a separate agent. For the vehicles, the virtual state is a desired time till arrival (DTTA), also denoted by $\tau_i(k)$, but for $i \in \mathcal{V}$. The DTTA evolution of the vehicles is governed by the same dynamics as that of the traffic light's DTTS, which is given by (4.1). In the case of the vehicles, the bounds in (4.2) are chosen to maintain feasible arrival times and they are expressed in subsection 4.2.3.2.

4.2.1.4 Problem Statement

Given the above model, the problem at hand consists of the following. First, we must define the coordinating control input $u_i(k)$ that drives the DTTA and DTTS of the agents to a suitable configuration where vehicles are planning to arrive during green windows. Second, we must define the arrival time bounds to ensure the agents can actually achieve the planned timings. Finally, we must define the acceleration control input $a_i(k)$ of each vehicle to place it at the intersection at the planned arrival time while maintaining safety and feasibility constraint of (4.4) and (4.5).

4.2.2 Gradient-Based Multi-agent Control

Multi-agent systems can be controlled by defining pairwise, positive semi-definite potential functions between connected agents, called edge tension functions (ETFs), that are locally defined with respect to the states of the ego agent and its neighbors. The agents then set their control input along the direction of the negative gradient of their total potential with respect to their respective states. This approach is commonly used in navigation and formation control problems [61]. The driving input $u_i(k)$ of (4.1) is then given by the negative gradient of a potential function defined by the local ETFs V_{ij}:

$$u_i(k) = \sum_{j \in \mathcal{N}_i} -\frac{\partial}{\partial \tau_i} V_{ij}(\tau_i, \tau_j) \tag{4.7}$$

where $\mathcal{N}_i$ is the set of neighbors of agent i in the multi-agent network. To drive the arrival and switching times of the agents we choose a logistic function as our ETF V_{ij}:

$$V_{ij} = \frac{1}{1 + e^{-\kappa_{ij}(\tau_i - \tau_j)}} \tag{4.8}$$

The logistic function is a suitable candidate for two main reasons. First, by changing the growth-rate parameter κ_{ij} from positive to negative, we change the direction of the force experienced by the ego agent from its neighbor. As such we can have interactions that accelerate or decelerate a given agent based on the relative DTTA/DTTS of the pair of agents. Second, the logistic function flattens out away from its midpoint, which in (4.8) corresponds to the arrival/switching time of the ego agent's neighbor. This means that agents further apart in the planned time domain do not interact with each other strongly. As such, the proposed approach leads to local solutions for the coordination problem.

The sign of the rate parameter κ_{ij} in the above equation determines whether the DTTA/DTTS of agent i is hastened or delayed by its interaction with agent j. The magnitude of κ_{ij} determines the interval of planned time in which interactions are non-negligible, with larger values of κ_{ij} resulting in steeper and more localized transitions of the ETF. We choose this parameter accordingly to achieve the following:

- The DTTA of Vehicles are hastened by traffic light switches to red (to catch an earlier green window) and delayed otherwise (to catch a later green window). Given the localized nature of the ETF this "pulls" a vehicle's arrival time towards the closest green window.

- The DTTA of vehicles are delayed by preceding vehicles and hastened by following vehicles. This maintains headways between vehicles in a queue.

- Traffic light switching times are delayed by earlier switching times, and hastened by later ones.

To formally characterize the sign of κ_{ij} we first define the following set. For all $i \in \mathcal{V}$ (i.e., for all vehicle agents) let $\mathcal{G}_i \subseteq \mathcal{N}_i \cap \mathcal{L}$ be the set of traffic light agents, neighbors of vehicle i, that correspond to a switch to green for i. Then κ_{ij}, and specifically its sign, is defined in Table 4.1.

		Agent j			
		$j \in \mathcal{V}$		$j \in \mathcal{L}$	
Agent i	$i \in \mathcal{V}$	$s_j < s_i$	$\kappa_{ij} = \kappa_1$	$j \in \mathcal{G}_i$	$\kappa_{ij} = \kappa_3$
		$s_j > s_i$	$\kappa_{ij} = -\kappa_2$	$j \notin \mathcal{G}_i$	$\kappa_{ij} = -\kappa_4$
	$i \in \mathcal{L}$	$i \in \mathcal{G}_j$	$\kappa_{ij} = \kappa_5$	$\tau_i < \tau_j$	$\kappa_{ij} = \kappa_7$
		$i \notin \mathcal{G}_j$	$\kappa_{ij} = -\kappa_6$	$\tau_i > \tau_j$	$\kappa_{ij} = -\kappa_8$

Table 4.1: Sign of ETF growth or decay rate κ_{ij} based on the type of agents i and j and their relative timing or position

In Table 4.1, κ_ℓ, $\ell = 1, 2, \ldots$, are the values of the growth or decay rate which determine the steepness of the transition in the logistic function. Lower values of κ_ℓ correspond to wider intervals of interaction between agents. We define the interval of interaction $\Delta\tau_\alpha$, as the interval around τ_j beyond which the ETF is smaller than α or larger than $1 - \alpha$, where α is a user-defined parameter that indicates what is considered a non-neglible interaction (we can use $\alpha = 0.01$). From (4.8), we can derive the relationship between the growth/decay rate κ_{ij} and the interaction interval $\Delta\tau_\alpha$:

$$\kappa_{ij} = \frac{1}{\Delta\tau_\alpha} \ln \frac{\alpha}{1 - \alpha} \tag{4.9}$$

To illustrate how the use of the logistic function as an ETF drives the DTTA/DTTS of the agents to a desirable configuration, we show an example scenario in Fig. 4.2. There are two vehicles approaching a junction from two conflicting directions and there is a single planned traffic light switch in the considered time horizon. In the top plot we see a snapshot of the current configuration at at particular time instant where the black dot represents the DTTS of the light, while the blue and red dots represent the DTTA of each vehicle. At this particular time instant the red vehicle plans to arrive in a green window and the blue vehicle plans to arrive in a red window. Given these relative timings, from Table 4.1 we can then find the parameters that define whether the ETFs between the agents have positive or negative gradients. From the top, the second, third, and bottom plots show the ETFs for the light, the blue vehicle, and the red vehicle, respectively. Since there are no other vehicles on each road, the blue and red vehicles are only influenced by the light, so their ETFs are given by a single line. The light, or black agent, is communicating with both vehicles, so its ETF is the sum of two curves (in black). Looking at the shape of the ETFs, the gradient-descent control law (4.7) implies that the black and red dot are pushed to the right and the blue dot is pushed to the left. In fact, for the light, we can see the aggregate effect of both edge tensions with the vehicle neighbors produces a valley that the DTTS will fall into through gradient descent control. As the agents coordinate, once that configuration is reached, both vehicles will be planning to arrive within a green window.

Remark In the previous chapter, timing coordination occurs through the Kuramoto model. However, as we mention in the introduction of this chapter, a Kuramoto system can be thought of as a gradient-based system. The corresponding ETF would be:

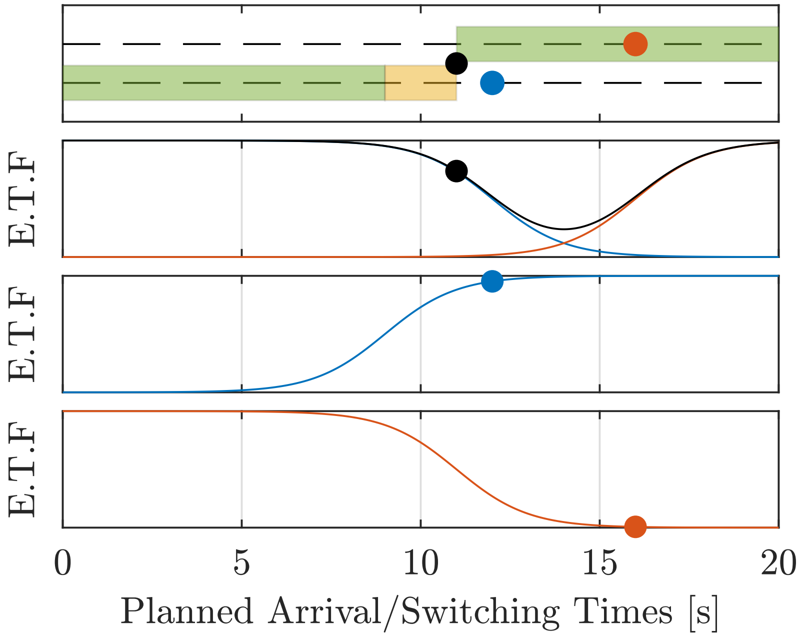


Figure 4.2: Illustrative example of proposed Edge Tension Functions (ETF)

$$V_{ij} = \omega_i \theta_i + \sin^2\left(\frac{\theta_j - \theta_i}{2}\right) \tag{4.10}$$

Given this similarity, we note how the proposed framework of this dissertation, can admit different types of coordinating strategies by encoding them into different ETFs.

4.2.3 Longitudinal Vehicle Control

While traffic lights can change their planned switching times instantaneously, vehicles must consider their dynamics when modifying their planned arrival times. A suitable speed profile must be followed for the vehicle to actually reach the intersection at the planned arrival time. Furthermore, because of constraints on speeds and accelerations of the vehicles, not all arrival times will be feasible for each of them. As such, a vehicle must constrain how much it modifies its DTTA in the consensus layer of the algorithm. In this subsection, we develop an optimal controller for planned arrival time tracking and we determine the corresponding bounds on planned arrival time that the optimal tracker imposes on the multi-agent consensus layer.

4.2.3.1 Optimal Arrival Time Tracking

The negotiated arrival time can be tracked by the vehicles using a simple optimal controller. At any time instant k, the nominal acceleration is the outcome of the following optimal control problem.

$$a_i^*(t) = \underset{a_i(\xi=0)}{\text{argmin}} \int_0^{\tau_i(t)} \frac{1}{2} a_i(\xi)^2 d\xi \tag{4.11}$$

Subject to:

$$\begin{aligned} \dot{s}_i(\xi) &= -v_i(\xi) \\ \dot{v}_i(\xi) &= a_i(\xi) \\ s_i(\xi = \tau_i(t)) &= 0 \\ s_i(\xi = 0) &= s_i(k) \\ v_i(\xi = 0) &= v_i(k) \end{aligned} \tag{4.12}$$

Following the work presented in [62], the above optimization problem can be solved analytically by defining co-states λ_i^s and λ_i^v, constructing the Hamiltonian, and imposing necessary conditions for optimality.

The Hamiltonian is:

$$H = \frac{1}{2}a_i(\xi)^2 - \lambda_i^s(\xi)v_i(\xi) + \lambda_i^v(\xi)a_i(\xi) \quad \forall \xi \in [0, \tau_i] \tag{4.13}$$

From the input stationarity condition we have that the optimal acceleration a_i^* follows:

$$\begin{gathered} \frac{\partial}{\partial a_i}H = a_i^*(\xi) + \lambda_i^v(\xi) = 0 \\ \therefore a_i^*(\xi) = -\lambda_i^v(\xi) \end{gathered} \tag{4.14}$$

From the co-state equations we can solve for the co-states in terms of integration constants c_1 and c_2:

$$\begin{aligned} &\dot{\lambda_i^s}(\xi) = -\frac{\partial}{\partial s_i}H = 0 \\ &\dot{\lambda_i^v}(\xi) = -\frac{\partial}{\partial v_i}H = \lambda_i^s \\ &\therefore \lambda_i^s(\xi) = -c1 \\ &\therefore \lambda_i^v(\xi) = -c_1\xi - c_2 \quad \forall \xi \in [0, \tau_i] \end{aligned} \tag{4.15}$$

It follows from (4.14) and (4.15), as well as from further integration of acceleration and position, that the optimal trajectories are:

$$\begin{aligned} a_i^*(\xi) &= c_1\xi + c_2 \\ v_i^*(\xi) &= \frac{1}{2}c_1\xi^2 + c_2\xi + c_3 \\ s_i^*(\xi) &= -\frac{1}{6}c_1\xi^3 - \frac{1}{2}c_2\xi^2 - c_3\xi + c_4 \end{aligned} \tag{4.16}$$

We can solve for the integration constants by enforcing the boundary conditions: $s_i^*(0) = s_i(t)$, $v_i^*(0) = s_i(t)$, $s_i^*(\tau_i(t)) = 0$ and $-\lambda_i^v(\tau_i(t)) = 0$.

$$\begin{aligned} c_1 &= 3\frac{c_3\tau_i(t) - c_4}{\tau_i(t)^3} \\ c_2 &= -c_1\tau_i(t) \\ c_3 &= v_i(t) \\ c_4 &= s_i(t) \end{aligned} \tag{4.17}$$

Finally, we can get the nominal acceleration input as the value of a_i^* at $\xi = 0$, which from (4.16) and (4.17) yields :

$$a_i^*(t) = 3\frac{s_i(t) - v_i(t)\tau_i(t)}{\tau_i(t)^2} \tag{4.18}$$

4.2.3.2 Additional Constraints

The above optimal control problem does not consider the constraints (4.4) and (4.5). We can guarantee that these constraints are satisfied by imposing suitable bounds $\tau_{i,min}(k)$ and $\tau_{i,max}(k)$, and by saturating the output of the optimal controller according to (4.5).

First, let's consider how bounds on DTTA can guarantee that the constraints (4.4) are satisfied. From the structure of the solution to the unconstrained problem of the previous section, we can derive an a priori range of values of DTTA $\tau_i(k)$ for which acceleration and speed constraints (4.4) are satisfied. In [62] a detailed derivation of this range is presented. Given the current state of the vehicle and its bounds on speed and acceleration, the minimum and maximum DTTA are:

$$\begin{aligned}
\tau_{i,\min}(k) &= \max\{\frac{3s_i(k)}{v_i(k)+2v_{max}}, \\
&\quad \frac{-3v_i(k)+\sqrt{9v_i(k)^2+12a_{min}s_i(k)}}{2a_{min}}\} \\
\tau_{i,\max}(k) &= \min\{\frac{3s_i(k)}{v_i(k)+2v_{min}}, \\
&\quad \frac{-3v_i(k)+\sqrt{9v_i(k)^2+12a_{max}s_i(k)}}{2a_{max}}\}
\end{aligned} \tag{4.19}$$

We use these values as the bounds of the saturation function in (4.2). By doing this, the solution to optimization problem (4.11) will adhere to constraints (4.4).

Finally, the ideal acceleration from (4.11) is compared to the acceleration limit corresponding to safe inter-vehicle spacing in (4.5). If the ideal acceleration is larger, the vehicle is forced to apply the safe vehicle acceleration from (4.5). For the model of $v_{i,b}$ in (4.5), we use the Gipps car following model [52] which is designed to maintain a safe headway from the preceding vehicle. When this acceleration is used, collision safety-related constraints override the solution to the vehicle speed trajectory optimization problem, and vehicle arrival at the upcoming intersection is delayed. Since coordination happens at every time step, this delay is eventually

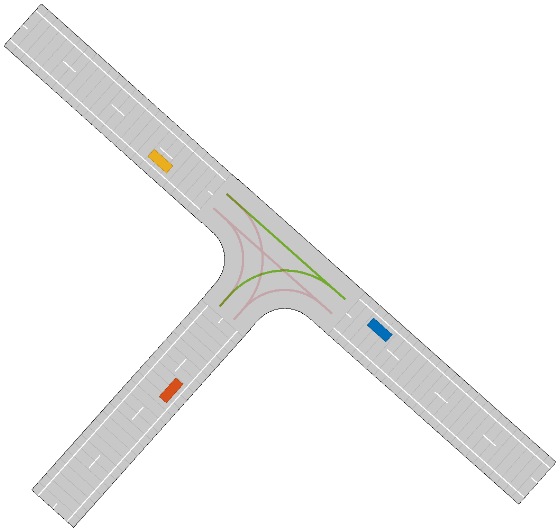


Figure 4.3: Example intersection with three lanes feeding the junction and a traffic controller

taken into account in the coordination layer.

4.3 Simulation Results

To showcase the strengths of the proposed strategy we simulate it under three different scenarios. The first scenario is a simple two-way intersection scenario, where the goal is to compare the performance with that of fixed time traffic lights and human driven vehicles for different saturation ratios. The second considers a three-way T-Junction scenario where the goal is to consider multiple levels of CAV penetration rates to show how the strategy has the flexibility to integrate with legacy technology. Finally, we consider a four-f way intersection, where we simulate changes and asymmetries in the traffic volume to showcase the ability of the approach to adapt to changing traffic conditions.

4.3.1 Simulation Parameters

The parameters κ_ℓ for our logistic functions are chosen according to a desired interaction interval between agents as discussed in the previous section. The vehicle-to-vehicle parameters κ_1 and κ_2 are set to 3. The vehicle-to-light parameters κ_3 and κ_4 are set to 1. The light-to-vehicle parameters κ_5 and κ_6 are set to 2. Finally the light-to-light parameters κ_7 and κ_8 are set to 2. For the minimum green window $G_{\min,p}$ we use 5 seconds and for the look-ahead time horizon T_H we use 30 seconds. As such, the traffic light adds a new DTTS agent switch, with $\tau_i = 30s$, when its latest DTTS reaches 25s.

4.3.2 Baseline

For the baseline simulation we consider human drivers plus fixed time traffic lights at the intersection. The human drivers are modeled using the Gipps car following model. The timing of the traffic light is determined using Webster's method [63].

4.3.3 Measures of Effectiveness

We analyze the results of the simulations based on two metrics: fuel consumption and delay. Fuel consumption, in grams per kilometer, is computed for each vehicle using longitudinal dynamics and an engine map for passenger cars. The delay, in seconds, corresponds to the difference between the time the vehicle stays in the network and the time it would have taken had it travelled at the desired speed. We set the desired speed for all vehicles to be $10m/s$ for all simulations. Delay and

Saturation Ratio:	0.644	0.75	0.84	0.92	0.987
Fuel Consumption:	-12.61%	-9.40%	-7.7%	-7.1%	3.08%
Delay:	-22.15%	-22.81%	-27.1%	-30.7%	-13.45%

Table 4.2: Changes in average fuel consumption and delay, from Baseline to Coordinated strategy, for Two-Way Junction simulation

fuel consumption are then averaged over all vehicles.

4.3.4 Case Study 1: Two-Way Junction and Saturation Rate Study

The first scenario replicates the configuration studied in [63]. It consists of one intersection of two, singled-lane, one way roads. Assuming a saturation flow rate of 1800 vehicles/hr/lane, we simulate the scenario for varying levels of saturation ratio, with balanced flow rates in each incoming lane. For example, a saturation ratio of 0.644 corresponds to a balanced input flow rate of 580 vehicles/hr in each lane. The simulation is run for 30 minutes.

In Table 4.2 we show the obtained changes in average fuel consumption and delay when comparing the baseline to the proposed strategy. We can see that our approach saves fuel and time for all but the most saturated scenario, where delay is reduced, but fuel consumption is not. In [63], comparisons between a similar baseline and different adaptive traffic light schemes for this scenario show savings in delay of 40% to 50%. This indicates that, while our approach has delay benefits as is, further gains can potentially be achieved, perhaps through better tuning of parameters.

4.3.5 Case Study 2: Three-Way Junction and CAV Penetration Study

In this study we simulate our approach for a three-way scenario (see Fig. 4.3), with varying numbers of human-driven cars compared to CAVs following the proposed strategy. For this simulation we assume that human-driven vehicles do not keep an arrival time state that is negotiated among other vehicles and the traffic light. However, we do assume that controlled agents, both smart cars and traffic lights, can sense the positions and velocities of human-driven vehicles. The connected agents then estimate the desired arrival time of unconnected vehicles $v_{i,a}$ using the free flow Gipps model from (4.6) integrated forward in time.

Figure 4.4 shows the results of this connected vehicle penetration rate study. All simulations are performed using the same vehicle injection times, which corresponds to an arrival rate of 500 vehicles/hour, for a saturation ratio of 0.83. We can see that fuel consumption decreases steadily with the relative number of connected vehicles. The delay exhibits a more surprising behaviour, with increases in connectivity of the fleet not necessarily translating into less delay till roughly 50% penetration, when the trend reverses. This suggests the possibility that having more cars modify their arrival time along with the traffic light is only advantageous if some critical percentage of vehicles is doing so.

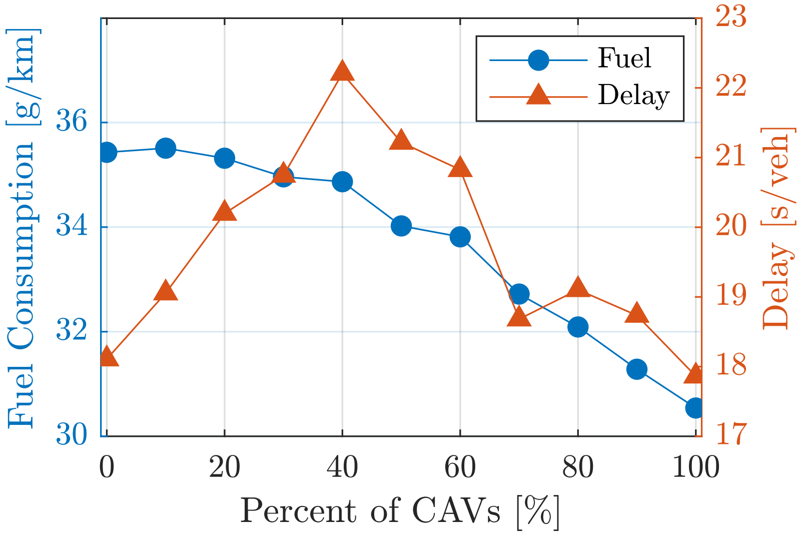


Figure 4.4: Fuel consumption and delay for T-Junction for different levels of CAV penetration rates

4.3.6 Case Study 3: Four-Way Junction and Varying Input Flow Rate Study

In this study, we consider a four-way intersection, with 8 incoming lanes grouped into 4 phases. Compatible left turns are grouped into phases 1 and 3, and straight and right turns in each direction (North-South and East-West) are grouped into phases 2 and 4. We simulate our strategy and the baseline for a saturation ratio of 0.667, which corresponds to a balanced flow of 300 vehicles/hr in each lane. However, we simulate two different varying input profiles for the duration of the simulation. The first, Arrival Profile I, starts with a balanced 300 vehicles/h in all lanes for 1000 seconds, then switches to an asymmetric 200 vehicles/hr in the North-South direction versus 400 vehicles/h in the East-West direction for the next 1000 seconds. Finally, it reverses the flow to 400 vehicles/hr in the North-South direction versus 200 in East-West direction for the last 1000 seconds. Arrival Profile

II follows the same pattern, starting at a balanced 300 vehicles/h, but then alternating between 500 versus 100 vehicles/s over the following 2000 seconds. The baseline traffic light timing is optimized for the first 1000 seconds of simulation, where the flow is balanced. The results are shown in Table **??**. We can see that in both cases our strategy saves fuel and delay.

Changing Traffic Volume:	Arrival Profile I	Arrival Profile II
	Average Fuel Consumtpion (g/km)	
Webster Fixed Opt.	39.4	46.2
Smart Lights, 100%CAV	32.9 (-16.5%)	40.9 (-11.5%)
	Average Delay (s/veh)	
Webster Fixed Opt.	34.5	70.5
Smart Lights, 100%CAV	23.0 (-33.3%)	53.7 (-23.9%)

Table 4.3: Results of Four-Way Intersection simulation with time varying arrival processes

4.4 Conclusion

In this chapter we propose a decentralized coordinating control strategy for both connected vehicles and traffic lights (agents) in an urban intersection. The framework consists of defining a potential energy field based on the logistic function that agents use to change their planned arrival/switching times. The potential function encodes suitable repulsion and attraction forces between these planned time states that drive the system into a desirable configuration. We define an optimal tracking controller for the vehicles to reach the intersection at the negotiated time and we use the characteristics of this controller to properly bound the coordinating upper layer of the algorithm. By using these bounds we guarantee that the negotiated arrival time can be tracked by the unconstrained optimal controller, which maintains the computational efficiency of the approach. We show that the proposed framework saves both time and fuel compared to fixed time traffic light control and that it does so for varying levels of autonomous vehicle penetration and saturation ratio.

Chapter 5: Vehicle and Traffic Light Control Through Gradient-Based Coordination and Control Barrier Function Safety Regulation

5.1 Introduction

5.1.1 Motivation

[1] So far, we have developed a control framework that coordinates vehicles and traffic lights, either by negotiating their planned timing directly or indirectly through their phases. The coordinated timing is then tracked by a trajectory planning controller. Safety and feasibility constraints have been considered in two different ways in the previous two chapters. In chapter 3, the Kuramoto-based strategy relied on the assumption that vehicles knew the entire planned trajectory of their leaders, this could happen either by communicating it, or by computing it themselves. Given the full trajectory of the leader, the trajectory planning algorithm would account for actuation bounds and safe car following requirements as explicit constraints of a numerical optimization problem. Notice, that this requires either heavy com-

[1] The work presented in this chapter is adapted from [64], under review in *Journal of Dynamic Systems, Measurement and Control , (2021)*

putational or connectivity capabilities. In chapter 4, the logistic-function based coordinator accounted for safety and feasibility constraints as an added saturation layer that overrides the acceleration command of vehicles. These ad-hoc constraints on the acceleration of the vehicles and on the planned arrival times are an interesting way of ensuring safety without increasing the complexity of the trajectory planner. However, they are implemented using the empirical Gipps car following model, and they do not account for the safety concerns regarding the traffic light changing its switching times in such a way that vehicles are forced to arrive on red. In this chapter we derive a safety regulator for both vehicles and traffic light agents within our coordinating control framework based on the use of control barrier functions.

The work in this chapter builds on existing research on safe-set nonlinear control methods to tackle the intelligent intersection management problem. Here, we integrate the use of the different approaches introduced in the previous chapter with control barrier functions to present a comprehensive control architecture that can flexibly coordinate the timing of both vehicles and traffic lights, while maintaining safety and feasibility.

Recall from Chapter 1 that a rich body of work explores the control of CAVs approaching unsignalized intersections. In [23, 25], comprehensive reviews of such approaches are presented. Like the approach presented in chapter 3, a common issue encountered by most of this work lies in the computational complexity and connectivity requirements of optimally tracking a given arrival time while considering other vehicles. Indeed, the work presented in [62] explores in detail the analytical solution to the arrival time tracking trajectory planning problem, and shows the difficulty of finding closed form solutions when considering all safety constraints. An alternative approach of particular relevance to the work presented here consists of generating a controller based primarily on safety considerations. For example, in [65] control barrier functions (CBFs) are used to synthesize a safety-critical decentralized control strategy for the vehicle merging problem, which is similar to the intelligent intersection management problem. The work is further extended in [66] to include a reference velocity that is explicitly related to coordinated desired arrival times.

5.1.2 Chapter Contributions

In this chapter we consider the decentralized control of both vehicles and traffic lights. Moreover, we are interested in the predictive control of these agents through the use of a timing state that governs when vehicles plan to arrive at the intersection and when the lights plan to switch. Indeed, such a control strategy continuously modifies not only the acceleration of the vehicles, but the planned state trajectories into the future of both the vehicles and the lights. As we discussed in the previous paragraphs, considering safety and feasibility constraints over a planning time horizon becomes computationally prohibitive. To address this issue, in this chapter, we make use of safe-sets and control barrier functions (CBFs) to generate a safety regulator that modifies a nominal controller that negotiates arrival times and plans acceleration trajectories (without explicitly considering safety and feasibility constraints). The main differences between this and previous contributions that use CBFs (i.e [65, 66]) are the inclusion of traffic lights and the consideration of an explicit state variable that governs the arrival times of the vehicles.

In the previous chapter we explore the coordinating and tracking control strategies that we use here as nominal controllers. There, coordination of arrival times is performed using gradient-based multi-agent methods, and arrival tracking is performed using analytical unconstrained optimal trajectory planning. In this previous work, car following safety constraints are implemented based on the Gipps model overriding the nominal acceleration commands, and actuation constraints are implemented by saturating the range of allowable arrival times of the vehicles. The interactions between different safety and feasibility constraints, and between traffic lights and vehicles were not formally analyzed. Thus, one of the main contributions of this work is the derivation of a safety regulating controller in conjunction with previously explored coordinating strategies.

5.1.3 Outline

The rest of the chapter is organized as follows. In Section 2 we describe the system and its dynamics. In Section 3 we give an overview of the proposed strategy. In Section 4 we define our nominal controllers. In Section 5 we derive the safety regulator which modifies the nominal control signals. Finally, in Section 6 we evaluate our controller in simulation.

5.2 Problem Formulation

The system consists of a multi-agent fleet of vehicles inside a control zone around a traffic intersection, plus the traffic light servicing the different flows at the intersection. The control zone is given by a maximum distance L from the intersection stop lines, along the lengths of the roads approaching it. In this section, we define the states, control inputs, and dynamics of these agents, as well as assumptions about their connectivity, autonomy and allowed behaviour.

We assume that once the vehicles enter the control zone, they can no longer change lanes. With regards to connectivity, we assume that all agents exchange their current state information with each other according to the following network topology. Vehicles in the same road can communicate with each other and with the upcoming traffic light. The traffic light, in turn, can exchange information with all vehicles approaching from all roads. As such, we can represent the network of interconnected agents as nodes in a bi-directional graph, and their connections as edges. We call $\mathcal{N}_i$ the set of neighbors of agent i.

5.2.1 Vehicle Dynamics

We consider the set $\mathcal{V}(t)$ of all vehicle agents currently approaching the intersection. We assume the i^{th} vehicle's state is given by its distance to the intersection s_i, its speed v_i, and its current planned desired time till arrival (DTTA) τ_i. The vehicle can control its acceleration a_i, and can change its DTTA by an added rate u_i. The dynamics of the vehicle are then:

$$\begin{aligned}\dot{s}_i(t) &= -v_i(t)\\ \dot{v}_i(t) &= a_i(t)\\ \dot{\tau}_i(t) &= -1 + u_i(t)\end{aligned} \tag{5.1}$$

The underlying assumption of using this longitudinal vehicle model is that lane keeping and powertrain controllers can keep the vehicle in its lane and meet the acceleration command used in Eqn. (5.1). We show the sate of three vehicles approaching a three way intersection in Fig. 5.1. The set $\mathcal{V}$ here contains agents 1, 2 and 3. In the spatial representation we can see the current distance to the intersection for each car, and in the timing domain we can see their planned arrival times.

5.2.2 Traffic Light Dynamics

The traffic light is modeled by its desired times till switching (DTTS), where each switch is considered its own independent agent for the purposes of the multi-agent control strategy described in the following section. We therefore consider the set $\mathcal{L}(t)$ of all upcoming planned DTTS, governed by the same dynamics as the DTTA of the vehicles:

$$\dot{\tau}_i(t) = -1 + u_i(t) \tag{5.2}$$

We assume here that the actuated lights follow a predetermined sequence of phases. If the intersection is organized around compatible movements grouped into P phases, we can index each phase from 1 to P, and assume that transitions happen in ascending order. Furthermore we define a minimum fixed yellow time T_{yellow} that separates the transition between phases. When the leading DTTS reaches T_{yellow}, the light switches from green to yellow for the current phase. When it reaches 0 the light switches from yellow to red for the current phase, and from red to green for the next phase in the sequence. We assume that the traffic light plans some time horizon T_{H} into the future, and that new planned traffic light switching events only spawn at T_{H}.

We illustrate the state of a traffic light controlling three flows in Fig. 5.1. In this example, agents 4, 5 and 6 belong to $\mathcal{L}(t)$. In the spatial domain, we can only see the current state of the light, which is servicing phase 1 at this time. In the timing domain, we can see the planned switches of the light over the coming T_{H} seconds.

5.3 Proposed Strategy

Given the above system of interconnected vehicle and traffic light agents, our objective is to synthesise a control strategy that generates the inputs a_i and u_i for all agents. Compared to human-driven and fixed-time traffic light control strategies, the controller should reduce the energy and delay incurred by having to stop at the light, while maintaining safety and feasibility constraints.

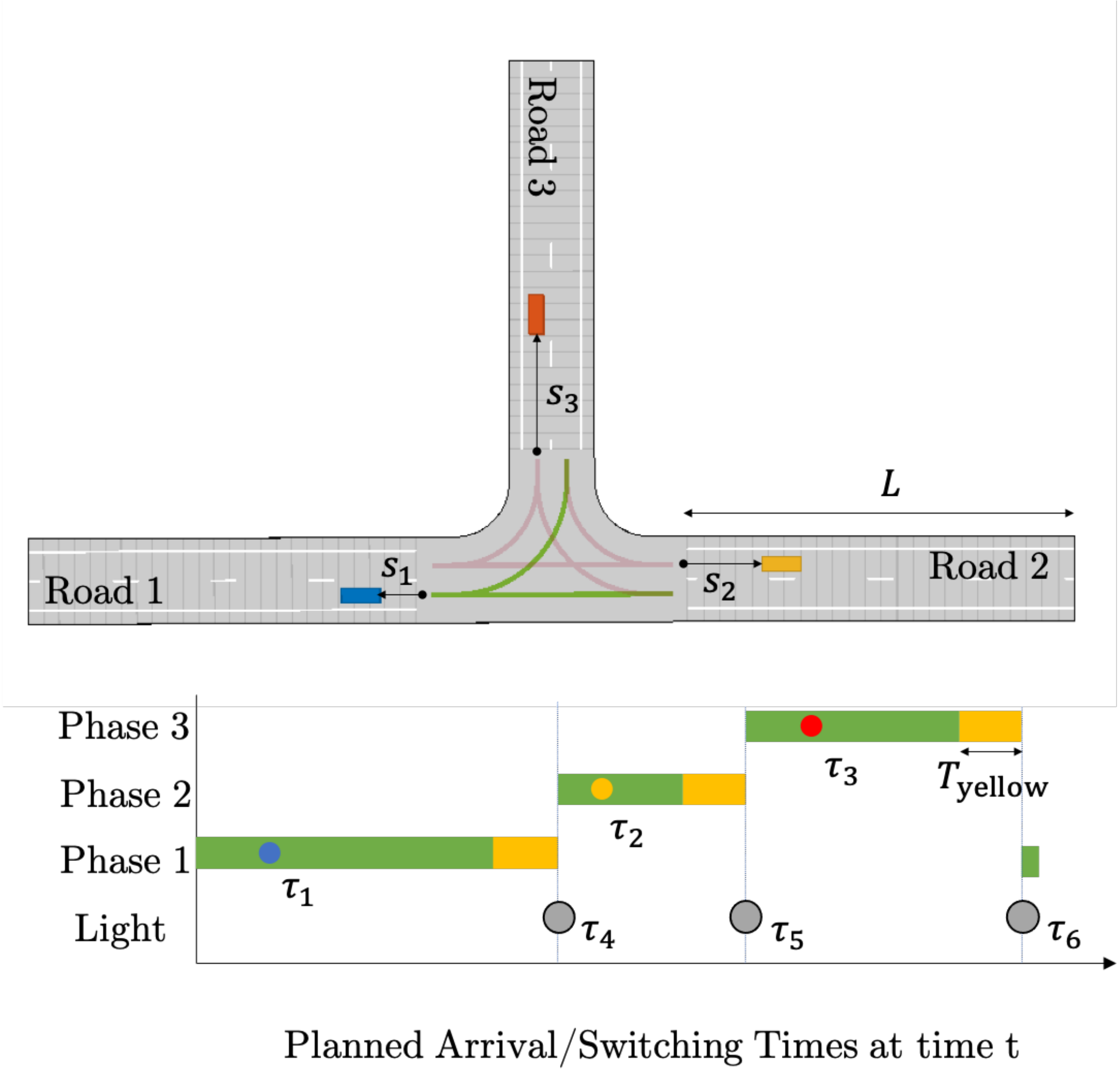


Figure 5.1: Vehicles and Traffic Light represented in the spatial and timing domain.

The proposed approach consists of having the agents continuously negotiate their timing by changing their DTTA/S. Negotiation is done through decentralized multi-agent methods; here, we use a gradient-based strategy. Then, a decentralized safety regulator for each agent considers constraints on inter-vehicle spacing, vehicles arriving on green periods, and consistent traffic light timings.

For each vehicle agent the proposed control strategy is illustrated in Fig. 5.2. A trajectory planning controller uses the current position, speed and DTTA of the vehicle to compute the nominal acceleration a_i^* that would place the vehicle at the intersection at the current DTTA, assuming the underlying safety constraints are satisfied. Simultaneously, a coordinating controller uses the communicated DTTA/S information from neighboring agents to determine the nominal change in timing u_i^*, assuming feasibility of the current DTTA is satisfied. Finally, a quadratic program (QP) controller minimizes the deviation from the nominal control input under safety and feasibility barrier certificate constraints.

For the traffic light the proposed control strategy is illustrated in Fig. 5.3. The approach is similar to that of the vehicles with a few key differences. The traffic light controller does not need a tracking algorithm, since there are no physical constraints on the rate at which it can change its DTTS. Furthermore, while each vehicle only controls its own DTTA, the traffic light controls all of its planned switching times together.

5.4 Nominal Controller

The nominal controller aims to coordinate the vehicles and lights by modifying their timings, and tracking the desired arrival times through trajectory planning optimal control, assuming that the underlying safety regulator will handle the safety and feasibility constraints. We denote the nominal control signals as u_i^* and a_i^*.

Since the main goal of this chapter is the derivation of the safety regulator, here we use the same coordinating potential function and the same tracking controller presented in the previous chapter. First, the ETFs V_{ij} between pairs of connected

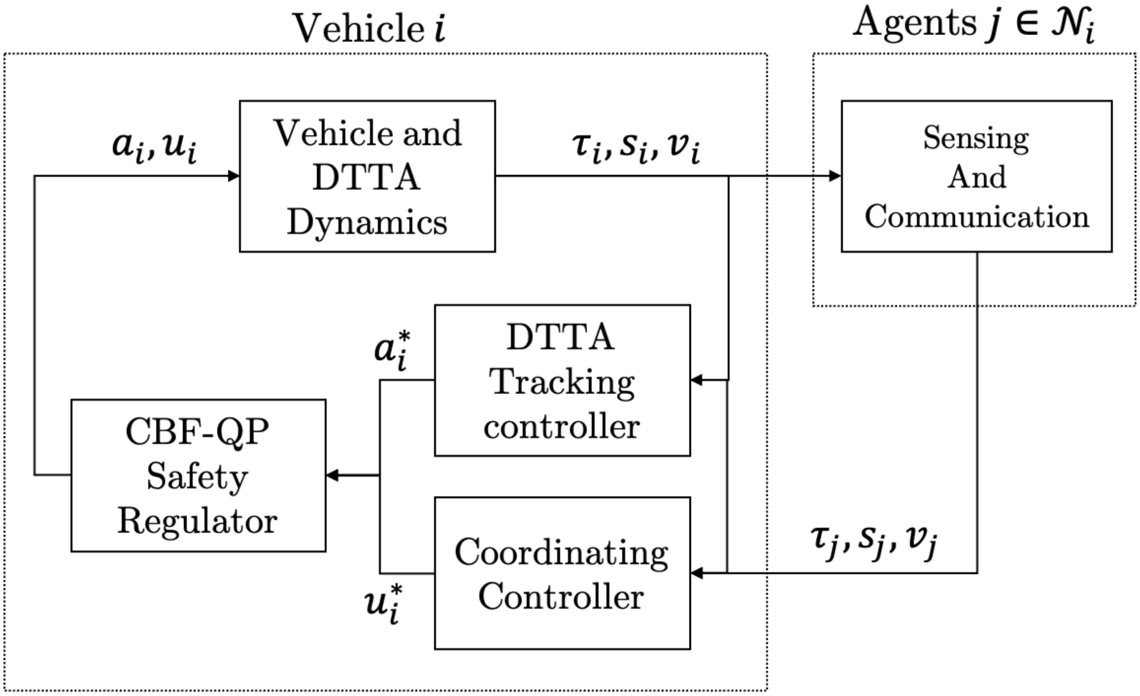


Figure 5.2: Control Architecture for a vehicle agent

agents i and j are defined using the logistic function:

$$V_{ij} = \frac{K_{ij}}{1 + e^{-\kappa_{ij}(\tau_i - \tau_j)}} \tag{5.3}$$

Second, the nominal acceleration control input that aims to place the vehicle at the intersection in $\tau_i(t)$ seconds is:

$$a_i^*(t) = 3\frac{s_i(t) - v_i(t)\tau_i(t)}{\tau_i(t)^2} \tag{5.4}$$

5.5 Safety Regulator

Given the nominal acceleration and coordinating input, the safety regulator will modify this input to maintain safety and feasibility constraints. To do this,

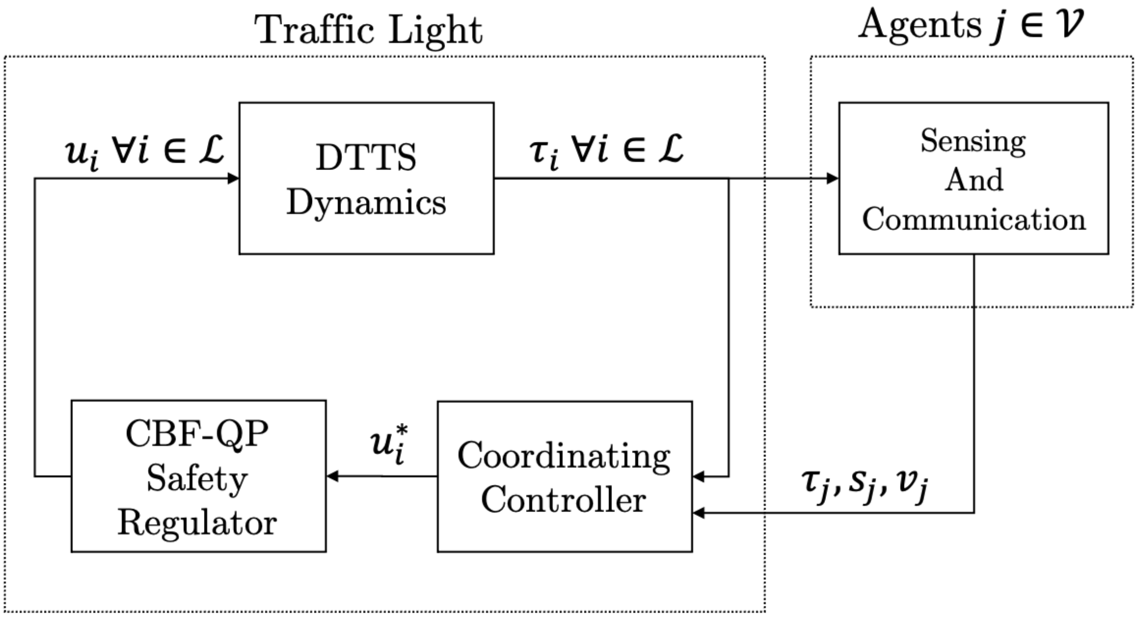

Traffic Light
Agents $j \in \mathcal{V}$
$u_i \ \forall i \in \mathcal{L}$
DTTS Dynamics
$\tau_i \ \forall i \in \mathcal{L}$
Sensing And Communication
CBF-QP Safety Regulator
u_i^*
Coordinating Controller
τ_j, s_j, v_j

Figure 5.3: Control Architecture for the traffic light

we make use of safety-critical control methods. Specifically, we seek the forward invariance of a safe set in the state space by the defining control barrier functions and enforcing their induced barrier certificates.

5.5.1 Background: Control Barrier Functions

The control barrier function method has been recently developed for safety-critical control applications, and a comprehensive introduction to the method can be found in [67]. Here, we recall the main definitions and theorems from [67] required in the formulation of our safety regulator.

We start by defining a control affine system, with states $\mathbf{x} \in D \subset \mathbb{R}^n$, input $\mathbf{u} \in U \subset \mathbb{R}^m$, and locally Lipschitz dynamics $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ and $\mathbf{g}$: $\mathbb{R}^n \to \mathbb{R}^{n \times m}$:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} \tag{5.5}$$

Definition 1 *A set $\mathcal{C}$ is forward invariant if for every initial condition $\mathbf{x}(0) = \mathbf{x_0} \in \mathcal{C}$, $\mathbf{x}(t) \in \mathcal{C}$ for all time t.*

Definition 2 *Set $\mathcal{C}$ is the superlevel set of a continuously differentiable function h: $D \subset \mathbb{R}^n \to \mathbb{R}$ if:*

$$\mathcal{C} = \{\mathbf{x} \in D \subset \mathbb{R}^n : h(\mathbf{x}) \geq 0\} \tag{5.6}$$

Definition 3 *An extended class $\mathcal{K}_\infty$ function is a strictly increasing function κ: $\mathbb{R} \to \mathbb{R}$ with $\kappa(0) = 0$.*

Definition 4 *The continuously differentiable function h: $D \subset \mathbb{R}^n \to \mathbb{R}$ is a control barrier function (CBF) if there exists an extended class $\mathcal{K}_\infty$ function κ such that for the system (5.5):*

$$\sup_{\mathbf{u} \in U} \left[L_f h(\mathbf{x}) + L_g h(\mathbf{x})^T \mathbf{u} + \kappa(h(\mathbf{x})) \right] \geq 0 \tag{5.7}$$

for all $x \in D$

The above definition uses the Lie derivative formalism to express the time derivative of the function h. Indeed, $\dot{h}(\mathbf{x}, \mathbf{u}) = L_f h(\mathbf{x}) + L_g h(\mathbf{x})^T \mathbf{u}$.

Intuitively, the condition says that, if there exists some input $\mathbf{u}$ and some extended class $\mathcal{K}_\infty$ function κ such that $\dot{h}(\mathbf{x}, \mathbf{u}) \geq -\kappa(h(\mathbf{x}))$, we call h a control barrier function. This is useful because applying any input $\mathbf{u}$ that makes

$\dot{h}(\mathbf{x},\mathbf{u}) \geq -\kappa(h(\mathbf{x}))$ will render the super-level set of h forward invariant. In fact, this constitutes the main theorem involving control barrier functions.

Theorem 1 (from [67]) *Let $\mathcal{C} \subset \mathbb{R}$ be the superlevel set of a continuously differentiable function h: $D \subset \mathbb{R}^n \to \mathbb{R}$. If h is a control barrier function and $\nabla_{\mathbf{x}} h(\mathbf{x}) \neq 0$ for all $\mathbf{x} \in \partial\mathcal{C}$, then any Lipschitz continuous controller $\mathbf{u}(\mathbf{x}) \in K_{cbf}(\mathbf{x})$ such that:*

$$K_{cbf}(\mathbf{x}) = \{\mathbf{u} \in U : L_f h(\mathbf{x}) + L_g h(\mathbf{x})^T \mathbf{u} + \kappa(h(\mathbf{x})) \geq 0\} \tag{5.8}$$

renders the set $\mathcal{C}$ forward invariant and asymptotically stable in D

The constraints on the input that define the set of allowable controllers are affine in the input. Given this property, we can synthesize a safe and computationally efficient controller using a linear or quadratic program, as shown in [68]. Given a nominal controller $\mathbf{u}^*(x)$ that is not guaranteed to be safe, but that aims to achieve some other control objectives, a quadratic programming (QP) problem can be designed to minimize the deviation from the nominal controller subject to the CBF constraint on the input:

$$\begin{gathered} \mathbf{u}(\mathbf{x}) = \underset{\mathbf{u} \in U}{\operatorname{argmin}} \frac{1}{2} ||\mathbf{u}(\mathbf{x}) - \mathbf{u}^*(\mathbf{x})||_2^2 \\ \text{Subject to:} \\ L_f h(\mathbf{x}) + L_g h(\mathbf{x})^T \mathbf{u} + \kappa(h(\mathbf{x})) \geq 0 \end{gathered} \tag{5.9}$$

The solution to this problem is shown to be Lipschitz continuous in [68]. When we have multiple constraints, or other constraints on the input, we can define candidate CBFs and formulate the above QP problem with one input constraint for each candidate CBF. In this case, the candidate CBFs would be valid CBFs if the QP problem has a solution for all $x \in D$.

In fact, there is another theorem that is helpful in generating multiple CBFs that are valid in the presence of each other, of other input constraints, or on higher relative degree dynamics. The idea consists of synthesizing CBFs from an initial candidate that may not be a valid CBF in the desired domain, plus a nominal controller that satisfies other input constraints. In [67], the candidate function is called a performance function and is denoted by ρ. The nominal controller is denoted by $\beta : D \to U$. The state of the system for any time time $t > 0$ is $\mathbf{x}_\beta(t, \mathbf{x})$, when it is initialized at $\mathbf{x}$ and input β is applied.

Theorem 2 (from [67]) *Let $\rho(\mathbf{x}) : D \to \mathbb{R}$ be a continuously differentiable performance function, $\beta(\mathbf{x}) : D \to U$ be a nominal controller such that $\dot{\mathbf{x}}(\mathbf{x}, \beta)$ is continuously differentiable, and $\mathbf{x}_\beta(t, \mathbf{x})$ be the state of the system at time t when it is initialized at $\mathbf{x}$ and input β is applied. If:*

$$h(\mathbf{x}) = \inf_{t \in [0,\infty)} \rho(\mathbf{x}_\beta(t, \mathbf{x})) \tag{5.10}$$

then:

- *h is a CBF;*
- $\mathcal{C} = \{\mathbf{x} \in D : h(\mathbf{x}) \geq 0\} \subseteq \{\mathbf{x} \in D : \rho(\mathbf{x}) \geq 0\}$
- $\beta(\mathbf{x}) \in K_{cbf}(\mathbf{x})$

5.5.2 Vehicle Safety and Feasibility Constraints

In the following section we make use of the above theoretical framework to synthesize a safety regulator for the vehicles. To do this, we introduce constraints on the states of the vehicles so that they remain in a safe configuration. The constraints are then translated into candidate control barrier functions, which are in turn transformed into barrier certificate constraints on the inputs.

Before defining the candidate barrier functions, we first establish the equivalence between our system (5.1), and the control affine system (5.5). For the vehicle agents, we use the following definitions for the state and input vectors and their dynamics:

$$
\begin{aligned}
\mathbf{x}_i &= [s_i, v_i, \tau_i]^T \\
\mathbf{u}_i &= [a_i, u_i]^T \\
\mathbf{f} &= [-v_i, 0, -1]^T \\
\mathbf{g} &= \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}
\end{aligned}
\tag{5.11}
$$

In the following, each new candidate CBF we consider is denoted by $h_q(\mathbf{x})$, for $q = 1, 2, .., 8$ (i.e. we define up to 8 different constraints on the states). Each h_q is associated with its own safe-set $\mathcal{C}_{i,q}$, and its induced allowable input space $U_{i,q}(\mathbf{x})$ as defined in Eq. (5.8). Since the barrier certificates are linear constraints on the input, we can organize them together into a single matrix inequality $A^{(i)}\mathbf{u}_i \leq \mathbf{b}^{(i)}$.

The rows of the inequality define $U_{i,q}(\mathbf{x})$:

$$U_{i,q}(\mathbf{x}_i) = \{\mathbf{u}_i \in \mathbb{R}^2 : A^{(i)}_{q,1} a_i + A^{(i)}_{q,2} u_i \leq b^{(i)}_q\} \tag{5.12}$$

Once all constraints are considered, any input $\mathbf{u}_i \in U_i(\mathbf{x}_i)$ will maintain the vehicle i inside of the safe set, where:

$$U_i(\mathbf{x}_i) = \bigcap_{q=0}^{Q} U_{i,q}(\mathbf{x}_i) \tag{5.13}$$

where Q is the total number of considered constraints.

Constraint 1[Acceleration bounds]

The first feasibility assumption we make for the vehicles is that their acceleration is constrained.

$$a_{\min,i} \leq a_i \leq a_{\max,i} \tag{5.14}$$

In practice, the values of $a_{\min,i}$ and $a_{\max,i}$ can be derived from the limitations of the vehicle's powertrain and the road.

Note that the acceleration constraints are not constraints on the states, but on the input, as such no CBF is associated with them. They define an initial admissible input set $U_{i,0} \subset \mathbb{R}^2$

$$U_{i,0} = \{[a_i, u_i]^T \in \mathbb{R}^2 : a_{\min,i} \leq a_i \leq a_{\max,i}, u_i \in \mathbb{R}\} \tag{5.15}$$

Constraints 1-2[Speed Limitations] Vehicles are constrained by the speed limits of the road, so that:

$$v_{\min} \leq v_i \leq v_{\max} \tag{5.16}$$

These constraints can be directly turned into candidate control barrier functions

$$h_1(\mathbf{x}_i) = v_i - v_{\min} \tag{5.17}$$

$$h_2(\mathbf{x}_i) = v_{\max} - v_i \tag{5.18}$$

To compute the corresponding barrier certificates, we choose a linear extended class $\mathcal{K}_\infty$ function $\kappa(h_p) = \gamma_{i,p} h_p$, where a different value of $\gamma > 0$ can be used for each CBF:

$$a_i + \gamma_{i,1}(v_i - v_{\min}) \geq 0 \tag{5.19}$$

$$-a_i + \gamma_{i,2}(v_{\max} - v_i) \geq 0 \tag{5.20}$$

Using the notation of Eq. (5.12) we define the coefficients of the matrix inequality:

$$
\begin{aligned}
A^{(i)}_{1,1} &= -1 \\
A^{(i)}_{1,2} &= 0 \\
b^{(i)}_{1} &= \gamma_{i,1}(v_i - v_{\min})
\end{aligned}
\tag{5.21}
$$

$$
\begin{aligned}
A^{(i)}_{2,1} &= 1 \\
A^{(i)}_{2,2} &= 0 \\
b^{(i)}_{2} &= \gamma_{i,2}(v_{\max} - v_i)
\end{aligned}
\tag{5.22}
$$

Constraints 3-6[Arrival Time Feasibility]:

The next constraints we consider limit the range of possible desired times till arrival (DTTAs) τ_i to those that are consistent with the vehicle's speed and acceleration limitations. In fact, we will limit the DTTAs to those whose nominal acceleration a_i^* and speed v_i^* trajectories (eq. (4.16)) do not violate acceleration and speed constraints (eqs. (5.14) and (5.16)).

From Eq. (4.16) we can see that the nominal trajectories for acceleration and speed are affine and quadratic functions of time respectively. What is more, for $t < \xi < t + \tau_i$, acceleration does not change signs; it starts at its extremum at $\xi = t$, then either increases or decreases until $a_i^*(\xi = t + \tau_i) = 0$. It follows that maximum or minimum speed occurs at $\xi = \tau_i$. From this analysis we can find corresponding constraints on the state space:

$$
a_{\min,i} \le a_i^*(\xi = t) \le a_{\max,i} \tag{5.23}
$$

$$v_{\min} \leq v_i^*(\xi = t + \tau_i) \leq v_{\max} \tag{5.24}$$

Using (4.16) and (4.17), we can rewrite these constraints as candidate CBFs:

$$h_3(\mathbf{x}_i) = a_{\max,i} - 3\frac{s_i(t) - v_i(t)\tau_i(t)}{\tau_i(t)^2} \tag{5.25}$$

$$h_4(\mathbf{x}_i) = 3\frac{s_i(t) - v_i(t)\tau_i(t)}{\tau_i(t)^2} - a_{\min,i} \tag{5.26}$$

$$h_5(\mathbf{x}_i) = v_i(t) + 3\frac{s_i(t) - v_i(t)\tau_i(t)}{2\tau_i(t)} - v_{\min} \tag{5.27}$$

$$h_6(\mathbf{x}_i) = v_{\max} - v_i(t) - 3\frac{s_i(t) - v_i(t)\tau_i(t)}{2\tau_i(t)} \tag{5.28}$$

Taking time derivatives of h_q, we can compute the corresponding coefficients of Eq. (5.12) :

$$\begin{aligned} A_{3,1}^{(i)} &= \frac{3}{\tau}a_i \\ A_{3,2}^{(i)} &= \frac{3v_i}{\tau_i^2} - \frac{6(v_i\tau_i - s_i)}{\tau_i^3} \\ b_3^{(i)} &= -\frac{6(v_i\tau_i - s_i)}{\tau_i^3} + \gamma_{i,3}h_3(\mathbf{x}_i) \end{aligned} \tag{5.29}$$

$$\begin{aligned} A_{4,1}^{(i)} &= -\frac{3}{\tau}a_i \\ A_{4,2}^{(i)} &= -\frac{3v_i}{\tau_i^2} + \frac{6(v_i\tau_i - s_i)}{\tau_i^3} \\ b_4^{(i)} &= \frac{6(v_i\tau_i - s_i)}{\tau_i^3} + \gamma_{i,4}h_4(\mathbf{x}_i) \end{aligned} \tag{5.30}$$

$$\begin{aligned} A_{5,1}^{(i)} &= \frac{1}{2} \\ A_{5,2}^{(i)} &= \frac{3v_i}{2\tau_i} - \frac{3(v_i\tau_i - s_i)}{2\tau_i^2} \\ b_5^{(i)} &= -\frac{3(v_i\tau_i - s_i)}{2\tau_i^2} + \gamma_{i,5}h_5(\mathbf{x}_i) \end{aligned} \tag{5.31}$$

$$\begin{aligned} A_{6,1}^{(i)} &= -\frac{1}{2} \\ A_{6,2}^{(i)} &= -\frac{3v_i}{2\tau_i} + \frac{3(v_i\tau_i - s_i)}{2\tau_i^2} \\ b_6^{(i)} &= \frac{3(v_i\tau_i - s_i)}{2\tau_i^2} + \gamma_{i,6}h_6(\mathbf{x}_i) \end{aligned} \tag{5.32}$$

Constraint 7[Safe Car Following]: The next constraint we introduce considers the spacing between the vehicle and its leader. For notation, let $l(i) := j$ if j is the directly ahead of vehicle i in its lane. We can then write a spacing constraint that requires vehicles to maintain a minimum distance D_{min} between them:

$$s_i - s_{l(i)} \geq D_{\text{min}} \tag{5.33}$$

Remark: This is the first constraint to depend on the states of another agent. In centralized formulations this is not a problem, as the inputs of both agents can be modified by the regulator. With decentralized regulators, where each vehicle is in charge of its own safety, the corresponding barrier certificate can be “split” between the agents, and still achieve good performance without sacrificing a large area of the admissible input space. In our case, we take a more conservative approach, where we assume a worse case input for the other agent.

The corresponding candidate CBF is:

$$\rho_7(\mathbf{x}_i, \mathbf{x}_{l(i)}) = s_i - s_{l(i)} - D_{\min} \tag{5.34}$$

We denote the function using ρ_7 instead of h_7 because, unlike previous ones, this constraint presents a more apparent issue when transforming it directly into a candidate barrier function. Namely, the constraint is on position, and its time derivative will be input independent (i.e. $L_g\rho(\mathbf{x}_i) = 0$). This is not necessarily a problem as long as (5.7) is still satisfied in some domain $D_{i,7}$, with $C_{i,7} \subseteq D_{i,7}$. In this case, the certificate is:

$$\dot{\rho}_7(\mathbf{x}_i, \mathbf{x}_{l(i)}) = -v_i + v_{l(i)} \geq -\gamma_7\rho_7(\mathbf{x}_i, \mathbf{x}_{l(i)}) \tag{5.35}$$

However, at the boundary of $C_{i,7}$, $\rho_7(\mathbf{x}) = 0$, and the above inequality reduces to $v_{l(i)} > v_i$. This means that the largest possible domain where ρ_7 is a valid CBF will be $D_{i,7} = \{\mathbf{x}_i, \mathbf{x}_{l(i)} \in \mathbb{R}^3 : v_{l(i)} \geq v_i\}$. Intuitively this makes sense: if the vehicle's leader is travelling faster than itself and it is more than $D_{\min}$ distance away, the vehicle cannot possibly reduce its spacing without first overtaking its leader in terms of speed and entering a set where the function is no longer a valid CBF. Therefore, we need a different candidate CBF.

Given the above discussion, when $v_{l(i)} \geq v_i$ no barrier certificate is needed to ensure safe car following. For $v_{l(i)} \leq v_i$, we derive a barrier function by assuming a nominal controller $a_i = \beta_i(\mathbf{x}_i)$ that represents the maximum braking maneuvers for both vehicles. The maximum braking maneuver is given by the constraints on acceleration and the barrier certificate on minimum velocity. Assuming $v_{\min} = 0$, this is given by the largest of two values: $\beta_i(v_i) = \max(a_{\min,i}, -\gamma_{i,1} v_i)$. We can then integrate the system forward in time under this control action, and compute the value of ρ_7 into the future. We then define the new candidate barrier function as the infima in time of ρ_7, as explained in Th. 2.

We can characterize the trajectory of ρ_7 as it is integrated into the future by noticing that, since $0 \leq v_{l(i)} \leq v_i$, ρ_7 monotonically decreases until both vehicles are completely stopped. We can therefore find its minimum value by adding the braking distance of each vehicle to the initial value of ρ_7. The braking distance of vehicle i under β_i is based on whether or not it starts at a speed larger than $v_{i,T} = -\frac{a_{\min,i}}{\gamma_{i,1}}$, which is the speed at which the braking controller transitions from from being $a_{\min,i}$ to being $-\gamma_1 v_i$. After computing the appropriate braking distance for each vehicle, based on their current speeds relative to $v_{i,T}$ and $v_{l(i),T}$, we can define the new candidate CBF:

$$h_7(\mathbf{x}_i, \mathbf{x}_{l(i)}) = \begin{cases} \text{for} \quad v_i \geq v_{i,T} \wedge v_{l(i)} \geq v_{l(i),T} : \\ \quad \rho_7(\mathbf{x}_i, \mathbf{x}_{l(i)}) + \frac{v_i^2}{2a_{\min,i}} - \frac{v_{l(i)}^2}{2a_{\min,l(i)}} \\ \quad + \frac{a_{\min,i}}{2\gamma_1^2} - \frac{a_{\min,l(i)}}{2\gamma_{1,l(i)}^2} \\ \text{for} \quad v_i \geq v_{i,T} \wedge v_{l(i)} < v_{l(i),T} : \\ \quad \rho_7(\mathbf{x}_i, \mathbf{x}_{l(i)}) + \frac{v_i^2}{2a_{\min,i}} + \frac{a_{\min,i}}{2\gamma_1^2} + \frac{v_{l(i)}}{\gamma_{1,l(i)}} \\ \text{for} \quad v_i < v_{i,T} \wedge v_{l(i)} < v_{l(i),T} : \\ \quad \rho_7(\mathbf{x}_i, \mathbf{x}_{l(i)}) + \frac{v_i}{\gamma_1} + \frac{v_{l(i)}}{\gamma_{1,l(i)}} \end{cases} \tag{5.36}$$

where $v_{i,T}$ corresponds to the speed where the maximum deceleration switches from being $a_{\min,i}$ to being $-\gamma_1 v_i$. That is, $v_{i,T} = -\frac{a_{\min,i}}{\gamma_1}$.

Simply put, the above candidate CBF consists of the original CBF h_7 plus the maximum braking distance of each vehicle when applying the maximum deceleration profile given by the constraints on acceleration and speed. The maximum braking distance of each vehicle depends on its current velocity. It should be noted that the above CBF candidate is continuously differentiable with respect to the state; this can be verified by evaluating h_7 and its gradient from both sides at each potential discontinuity point.

Differentiating in time, and assuming the leader uses its allowed maximum deceleration we get the barrier certificate constraint on the input.

$$
\begin{aligned}
A_{7,1}^{(i)} &= \begin{cases} -\frac{v_i}{a_{\min,i}} \quad \text{if} \quad v_i \geq v_{i,T} \\ \\ \frac{1}{\gamma_{i,1}} \quad \text{if} \quad v_i < v_{i,T} \end{cases} \\
A_{7,2}^{(i)} &= 0 \\
b_7^{(i)} &= -v_i + \gamma_7 h_7(\mathbf{x}_i, \mathbf{x}_{l(i)})
\end{aligned} \tag{5.37}
$$

Remark: To get this constraint, we need to assume that the deceleration parameters of the leader $a_{\min,l(i)}$ and $\gamma_{1,l(i)}$ are known.

Constraint 8[Safe Crossing Constraint]: Finally we introduce a constraint that guarantees that the vehicle only enters the intersection when it is planning to arrive on green.

The idea consists of reproducing the previous barrier function h_7 but assuming the leader's distance to the intersection and speed are both zero, as if there was a vehicle stopped at the intersection stop line.

$$
h_8(\mathbf{x}_i) = \begin{cases} s_i - D_{\min} + \frac{v_i^2}{2a_{\min,i}} + \frac{a_{\min,i}}{2\gamma_1^2} & v_i \geq v_{i,T} \\ \\ s_i - D_{\min} + \frac{v_i}{\gamma_1} & v_i < v_{i,T} \end{cases} \tag{5.38}
$$

The induced admissible set of inputs is given by:

$$
\begin{aligned}
A_{8,1}^{(i)} &= \begin{cases} -\frac{v_i}{a_{\min,i}} \quad \text{if} \quad v_i \geq v_{i,T} \\ \\ \frac{1}{\gamma_{i,1}} \quad \text{if} \quad v_i < v_{i,T} \end{cases} \\
A_{8,2}^{(i)} &= 0 \\
b_8^{(i)} &= -v_i + \gamma_{i,8} h_8(\mathbf{x}_i)
\end{aligned}
\tag{5.39}
$$

The above constraint is only used when the vehicle is planning to arrive during a red period of time. Notice then that h_8 can only become negative when the vehicle is planning to arrive on green. When h_8 is negative, the vehicle will no longer delay its DTTA (i.e. $u_i \geq 0$), and the traffic light will ensure that the light's planning does change in such a way that the vehicle is then forced to arrive on red. This is explained in following sections.

Definition 5 *Let $G(i)$ be an indicator function that is 1 when vehicle i is planning to arrive on green, and 0 when it is planning to arrive on red.*

5.5.3 Vehicle QP Safety and Feasibility Regulator

We can now define the QP safety regulator for each vehicle i.

$$
\begin{aligned}
\mathbf{u}_i(\mathbf{x}_i,\mathbf{x}_j) =& \underset{a_i,u_i}{\operatorname{argmin}} \frac{1}{2}(a_i - a_i^*(\mathbf{x}_i))^2 + \frac{1}{2}(u_i - u_i^*(\mathbf{x}_i,\mathbf{x}_j))^2 \\
&\text{Subject to:} \\
&a_{\min,i} \leq a_i \leq a_{\max,i} \\
&A_{q,1}^{(i)} a_i + A_{q,2}^{(i)} u_i \leq b_q^{(i)} \quad \text{for} \quad q = 1, ..., 6 \\
&A_{7,1}^{(i)} a_i + A_{7,2}^{(i)} u_i \leq b_7^{(i)} \quad \text{if} \quad v_i > v_{l(i)} \\
&A_{8,1}^{(i)} a_i + A_{8,2}^{(i)} u_i \leq b_8^{(i)} \quad \text{if} \quad G(i) = 0 \\
&u_i \geq 0 \quad \text{if} \quad h_8(\mathbf{x}_i) \leq 0
\end{aligned}
\tag{5.40}
$$

5.5.4 Traffic Light Safety Constraints

For the traffic light we also establish constraints on the allowable rate with which the planned switching times can be changed based on candidate CBFs. We then formulate a safety regulating QP controller that modifies the nominal change in timing generated by the coordinating gradient-based control strategy.

While we consider each DTTS of the light as separate agent for the purposes of gradient-based coordination, in this section we derive a single QP safety regulator for all the DTTS of a single light. This is different from the regulator derived for the vehicles, which assumes only the acceleration and rate of change of τ_i are control variables. Here, u_i for all $i \in \mathcal{L}(t)$ are optimization variables of the QP regulator. We therefore define the vector state of the light $\mathbf{x}_\mathcal{L}$ and the vector input $\mathbf{u}_\mathcal{L}$ as a concatenation of all τ_i and u_i for all $i \in \mathcal{L}(t)$.

Constraint 9 Given that we are assuming the light follows a predetermined sequence of phases, and that transitions occur whenever the leading DTTS reaches $-T_{\text{yellow}}$, we can guarantee the safe operation of the traffic light by ensuring that the DTTS do not overtake each other and always stay T_{yellow} seconds apart in the planned timing domain.

$$h_9(\tau_i, \tau_j) = \tau_j - \tau_i - T_{\text{yellow}} \qquad \forall i, j \in \{i, j \in \mathcal{L}(t) : \tau_i < \tau_j\} \tag{5.41}$$

The barrier certificate input constraints are then:

$$u_j - u_i \geq -\gamma_{i,9} h_9(\tau_i, \tau_j) \qquad \forall i, j \in \{i, j \in \mathcal{L}(t) : \tau_i < \tau_j\} \tag{5.42}$$

Constraint 10 The second set of constraints on the DTTS ensures that once a vehicle can no longer stop at the upcoming intersection, the light does not change its plan in such a way that the vehicle arrives in red. This constraint is used once a vehicle's h_8 CBF is negative (which can only happen if the vehicle is planning to arrive on green). In other words, once the vehicle crosses a safe stopping threshold, the light cannot hasten or delay its switches and make the vehicle arrive on red.

$$h_{10}(\tau_i, \tau_j) = \begin{cases} \tau_i - \tau_j & \text{if} \quad \tau_i \geq \tau_j \\ \tau_j - \tau_i & \text{if} \quad \tau_i < \tau_j \end{cases}$$

$$\forall i \in \mathcal{L}(t) \quad \text{and}$$

$$\forall j \in \{j \in \mathcal{V}(t) : h_8(\mathbf{x}_j) \leq 0\} \quad (5.43)$$

Assuming that vehicle j uses $u_j = 0$, the barrier certificate input constraints are then:

$$\begin{cases} u_i \geq -\gamma_{i,10} h_{10}(\tau_i, \tau_j) & \text{if} \quad \tau_i \geq \tau_j \\ -u_i \geq -\gamma_{i,10} h_{10}(\tau_i, \tau_j) & \text{if} \quad \tau_i < \tau_j \end{cases}$$

$$\forall i \in \mathcal{L}(t) \quad \text{and}$$

$$\forall j \in \{j \in \mathcal{V}(t) : h_8(\mathbf{x}_j) \leq 0\} \quad (5.44)$$

5.5.5 Traffic Light QP Safety Regulator

We can now define the QP safety regulator controller for each traffic light.

$$\mathbf{u}_{\mathcal{L}}(\mathbf{x}_{\mathcal{L}}, \mathbf{x}_j) = \underset{\mathbf{u}_{\mathcal{L}}}{\operatorname{argmin}} \frac{1}{2}||\mathbf{u}_{\mathcal{L}} - \mathbf{u}^*_{\mathcal{L}}(\mathbf{x}_{\mathcal{L}}, \mathbf{x}_j)||^2$$

$$\text{Subject to:} \tag{5.45}$$

$$\text{Eqs. } (5.42)$$

$$\text{Eqs. } (5.44)$$

Remark: The above QP controller for the traffic light simply attempts to preserve the order of the DTTS. When some vehicles can no longer stop at the intersection, it also attempts to preserve the order of the DTTA of these vehicles relative to the DTTS of the light. We can see that there exists at least one feasible solution by noting that $u_i = 0 \quad \forall i \in \mathcal{L}$ satisfies all of the constraints. Indeed, if the agents start in a certain order, and they all set their inputs to 0, they should preserve that order, given their homogeneous affine dynamics.

5.5.6 Constraint Relaxations

As we have discussed previously, the QP-CBF formulation has been derived using candidate CBFs, which are only valid CBFs in the presence of each other if the QP problem has a feasible solution ($U_i \neq \emptyset$) for all points inside the safe set ($\mathbf{x}_i \in \mathcal{C}_i$). In this section, we consider the possibility of the set U_i being empty for the vehicles, and we introduce appropriate constraint relaxations to ensure that the QP always has a solution. Remark 3 in the previous subsection discusses how this is not necessary for the traffic light.

In Fig. 5.4 we show an example of the vehicle's constrained input space

$U_i(\mathbf{x}_i, \mathbf{x}_{l(i)})$, for a given value of $\mathbf{x}_i$ and $\mathbf{x}_{l(i)}$. The allowable input space is left uncolored, and it is the intersection of all the individual barrier certificates.

In the previous sections, we derive constraints 7 and 8 to be explicitly compatible with constraints 0 and 1 using Theorem 2, and we can show their compatibility with constraint 2 as this constraint cannot become active when the vehicle is braking as long as $v_{\max} > v_{\min}$. This implies that in Fig.5.4 the vertical region between $U_{i,1}$ and $U_{i,7}$ (and $U_{i,8}$) will always be non-empty. In other words, $U_{i,7}$ (and $U_{i,8}$) will always be to the right of $U_{i,1}$.

On the other hand, the derivation of constraints 3 through 6 does not consider any constraint interactions. As such, the polytope formed by these four lines might become empty, or it might not intersect with the vertical region formed by $U_{i,1}$ and $U_{i,7}$ (or $U_{i,8}$). Given this possibility, we introduce a constraint relaxation input $\delta_{i,q}$ in $U_{i,q}$, for $q = 3, 4, 5, 6$, as follows:

$$\begin{aligned} U_{i,q}(\mathbf{x}_i) =& \{\mathbf{u}_i \in \mathbb{R}^2 : A^{(i)}_{q,1} a_i + A^{(i)}_{q,2} u_i + \delta_{i,q} \leq b^{(i)}_q\} \\ & \forall q \in \{3, 4, 5, 6\} \end{aligned} \tag{5.46}$$

We can relax the vehicle's QP safety regulator defined in Eq.(5.40) by adding the $\delta_{i,q}$ as optimization variables, replacing the appropriate constraints by the ones defined above (Eq. (5.46)), and replacing the optimization objective by the following:

$$\frac{1}{2}(a_i - a_i^*(\mathbf{x}_i))^2 + \frac{1}{2}(u_i - u_i^*(\mathbf{x}_i, \mathbf{x}_j))^2 + R \sum_{q=3}^{6} \delta_{i,q}^2 \tag{5.47}$$

The relaxation weight R is chosen to be a very large number to heavily discourage the controller form violating the constraint.

Remark: In practice, there are other reasons to relax the constraints and heavily penalize their violation. First, in the field and in simulation this strategy will be implemented as a discrete-time controller where, near the boundary of the safe-set, the forward invariance of the set may not be guaranteed if the integration time step size is not small enough. Second, in real roads with human drivers, some of the assumptions about the behaviour of the leading traffic could be violated, unwillingly placing the vehicle outside its safe set.

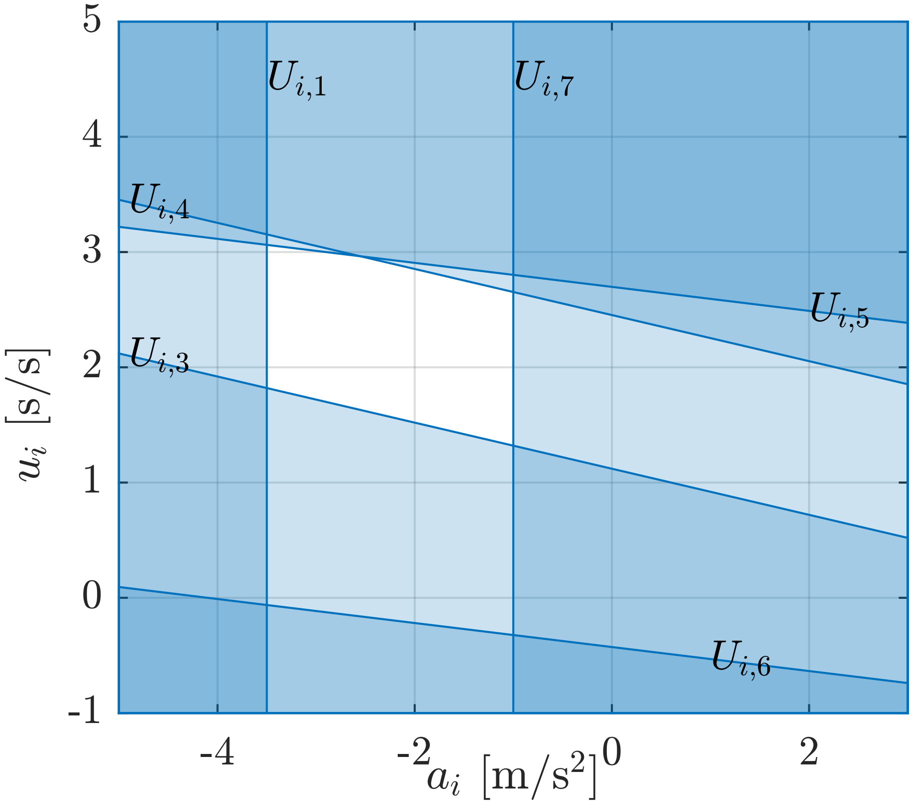


Figure 5.4: Example schematic of the allowable input set $U_i(\mathbf{x}_i, \mathbf{x}_{l(1)})$ for a given value of $\mathbf{x}_i$ and $\mathbf{x}_{l(i)}$.

5.6 Simulation Results

In this section, we evaluate our control strategy in simulation. We consider a three-way intersection controlled by a traffic light, the same configuration as shown in Fig. 5.1.

We assume that vehicles arrive in each lane according to a Poisson arrival process, given by an input flow rate in vehicles per hour. Arrivals are delayed to the extend that they would have the vehicle enter outside of the safe-set. For the "arrival" of traffic light switching events, if the light plans a time horizon T_{H} into the future, a new DTTS is added at $\tau_i = T_{\text{H}}$ when the last DTTS is some minimum time T_{min} into the planned timing domain.

Assuming a 2 second headway, the saturation flow rate of each lane is 1800 veh/hr. We simulate our proposed strategy and baseline at different ratios of the saturation flow rate using the parameters summarized in Table 5.1. For each saturation ratio we generate 10 realizations of the arrival process, and we use the same generated arrival times for both the baseline and the proposed strategy.

5.6.1 Baseline

The baseline strategy consists of human drivers and fixed time traffic lights at the intersection. The human drivers are modeled using the Gipps car following model [52], with a desired speed v_{des}. The timing of the traffic light is determined using Webster's method [63], based on the demand (i.e. arrival rate) at each road.

Parameter	Value	Units
L	100	m
$a_{\text{min},i}$	-5	m/s^2
$a_{\text{max},i}$	3	m/s^2
v_{min}	0	m/s
v_{max}	15	m/s
v_{des}	10	m/s
D_{min}	7	m
$\kappa_1, ...\kappa_8$	1	-
K_{ij}	10	-
$\gamma_{i,1}, \gamma_{i,2}$	5	1/s
$\gamma_{i,3}, ...\gamma_{i,10}$	1	-
R	10^6	-
T_{H}	5	s
T_{min}	5	s
T_{yellow}	2	s

Table 5.1: Parameter values used in simulation

5.6.2 Results

We can look at the savings in fuel and delay incurred by using our proposed strategy over the baseline. In Fig. 5.5 we show the average savings per vehicle in both delay and fuel respectively. We can see that the coordination strategy reliably saves energy and time at all saturation ratios, and that in general, at higher volumes the savings are more substantial.

Remark: Fuel consumption is calculated from acceleration using an engine map for passenger vehicles. Delay is calculated as the difference between the time it would have taken the vehicle to traverse the network at a speed v_{des} and the actual travel.

To understand these results we can first look at the time-space diagrams for one lanes in one of the simulations. In Fig. 5.6, we show the trajectories of all vehicles that traverse the one of the lanes for both the baseline and proposed strategy. We can graphically see that the coordinating strategy generates smoother trajectories and reduces stopped time, which explains the reduction in delay and fuel consumption.

From the savings results, we can also see the strategy saves significantly more in delay than it does in fuel consumption. We believe this happens because the control strategy sometimes accelerates vehicles in order to catch a green light, a behaviour that is not usually modelled by the Gipps car following model. We can see that this is the case in Fig. 5.7. Here we plot the distribution of speeds for all vehicles at all time steps for the 0.7 saturation rate simulation. We can see the the coordinating strategy not only has the vehicles stopped less often, but it sometimes sees the vehicles traveling at speeds higher than the desired speed.

5.7 Conclusion

The main objective of the work in this chapter is to create a safety regulating controller that takes the negotiating strategy presented in previous chapters, and modify it in order to ensure safe operation, while maintaining a computationally efficient trajectory planning nominal controller. The regulator is synthesized using control barrier certificates implemented through a quadratic program that aims to minimize deviations from the nominal coordinating control. The overall control strategy, include the coordination strategy of Chapter 4 and the safety regulator derived in this chapter, is tested in simulation and compared to traditional traffic light and human-driven behaviour to showcase the improvement in fuel consumption and delay.

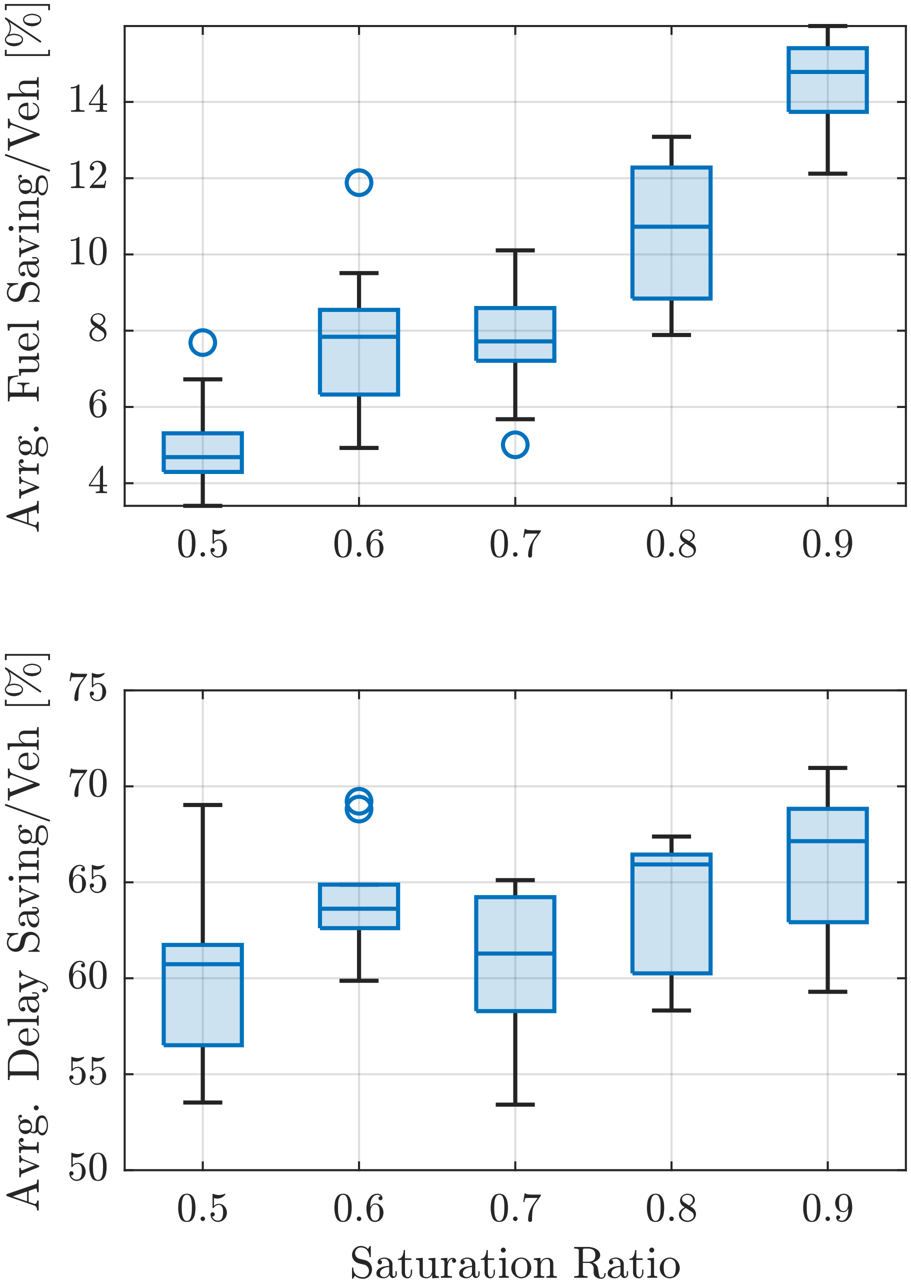


Figure 5.5: Average fuel savings per vehicle, for different saturation ratios

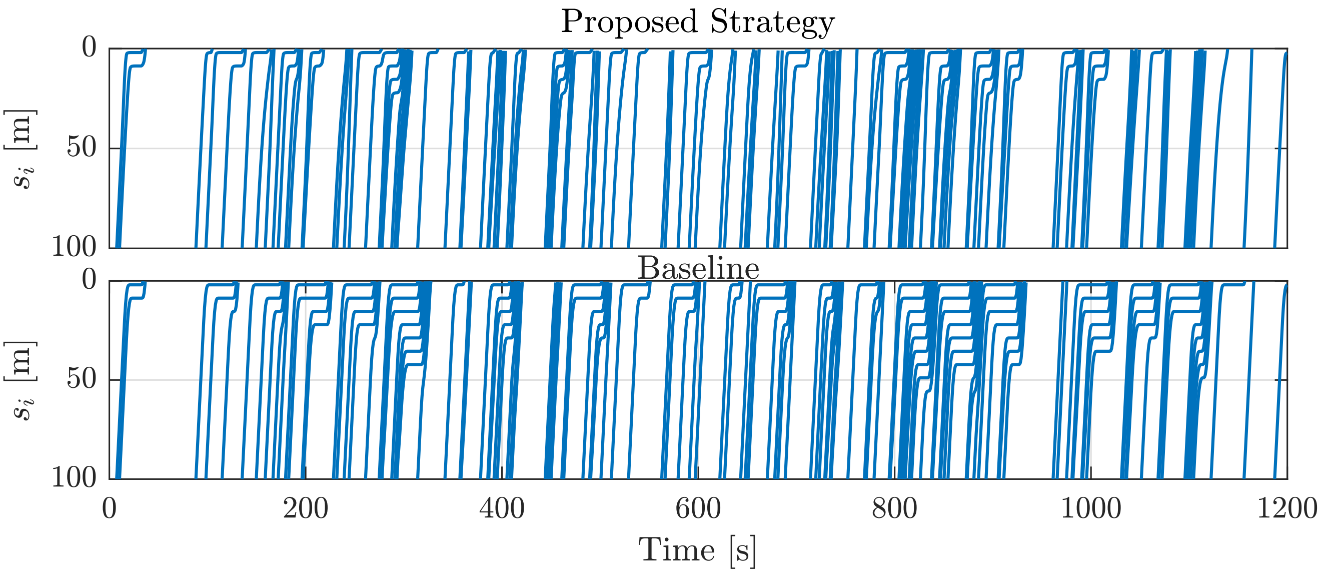


Figure 5.6: Proposed control architecture for vehicle agents

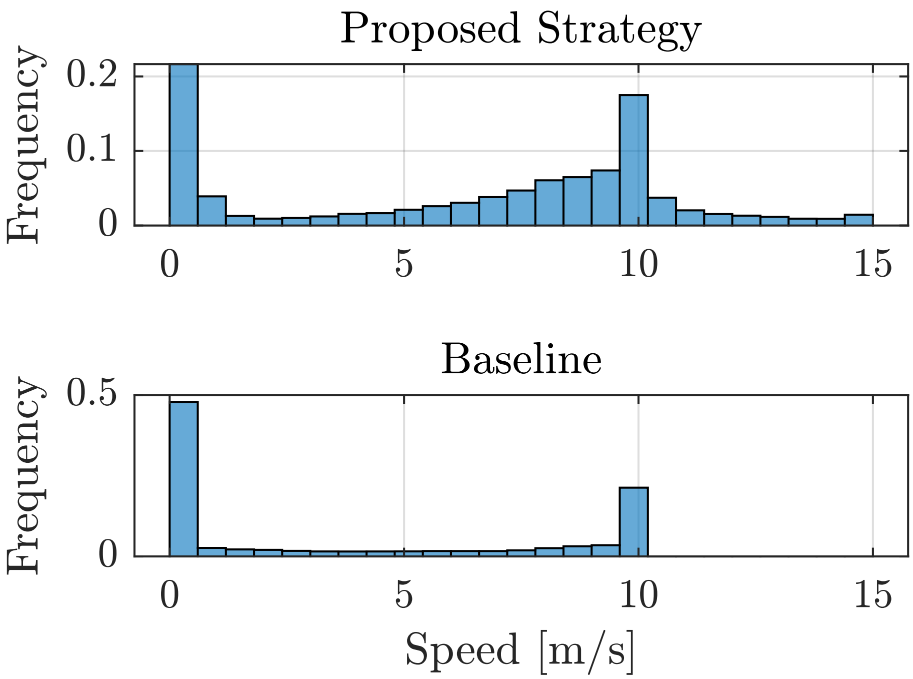


Figure 5.7: Proposed strategy and baseline distributions of speeds, for one simulation scenario at 0.7 saturation rate

Chapter 6: Conclusion

This dissertation proposes a decentralized coordination control framework for smart vehicles and traffic lights in urban intersections. A rich body of literature establishes the benefits of leveraging connectivity and automation to improve the performance of traffic networks. However, proposed approaches focus mostly on coordinating either only vehicles or only traffic lights. When both of these types of agents are considered in a single framework, approaches generally use optimization methods. In contrast, the framework presented in this dissertation uses gradient-based methods coupled with a control-barrier function safety regulator. Gradient-based methods have advantages in terms of scalability and computational simplicity. What is more, the use of safe-set methods to ensure safety allows the trajectory planning algorithm to remain computationally tractable and to not require excessive amounts of information from the leading traffic. Another interesting characteristic of the proposed framework is its flexibility. The approach is designed to accommodate different edge-tension functions, and can therefore work with different coordination strategies. We show both an approach based on the Kuramoto model for synchronization, and an approach based on logistic functions that hasten and delay the timing of neighboring agents.

Throughout the dissertation, the above framework is developed in 6 chapters, each progressively contributing to its final derivation. As such, the main body of this document can be summarized as follows:

- Chapter 1 presents a comprehensive review of the literature on the intelligent intersection management (IIM) problem. We summarize the different approaches other researchers have proposed to solve this problem, and we highlight the need for decentralized, scalable and efficient methods that include both smart vehicles and traffic lights.

- Chapter 2 explores our own motivating study into the IIM problem. The results from this chapter showcase the complexity of the trajectory planning problem in the presence of other vehicles and traffic lights, and the important role that coordination across adjacent intersections can play.

- Chapter 3 introduces our first iteration of the proposed framework in this dissertation: a multi-layer strategy where gradient-based coordination of a virtual variable that governs the agents' timing is followed by tracking that attempts to meet the planned timing. In this chapter we consider only CAVs, and we describe planned timings across multiple intersections using a novel definition of a vehicle's phase. The gradient-based strategy is given by the Kuramoto model for the synchronization of non-linear oscillators.

- Chapter 4 defines, in a similar framework, a new coordination strategy that includes both vehicles and traffic lights based on logistic function that define the interaction between agents.

- Chapter 5 derives a safety regulator for the proposed strategy that modifies the nominal controllers presented in previous chapters. As such, safety constraints do not need to be handled by the coordination or trajectory planning layers, which allows for more computational efficiency and looser connectivity requirements.

We have developed and validated a control framework that opens the door for productive future research. The approach we lay out it here lends itself to account for more complex traffic scenarios. Indeed, as we consider interactions between multiple intersections with traffic lights, we can combine the coordinating approaches of Chapters 3 and 4 to create ETFs between different types of agents at different intersections. In fact, the contributions of Chapter 5 allow us to explore a panacea of coordinating strategies through diverse designs of ETFs without needing to worry about safety considerations at the level of timing coordination. As such, we could even explore coordinating strategies through the use of machine learning approaches in the design of the ETFs that coordinate the timing of agents.

Bibliography

[1] David Schrank, Bill Eisele, and Tim Lomax. Urban Mobility Report 2019. page 50, 2019.

[2] Ardalan Vahidi and Antonio Sciarretta. Energy saving potentials of connected and automated vehicles. *Transportation Research Part C: Emerging Technologies*, 95:822–843, 2018.

[3] P. B. Hunt, D. I. Robertson, R. D. Bretherton, and R. I. Winton. Scoot - a Traffic Responsive Method of Coordinating Signals. *TRRL Laboratory Report (Transport and Road Research Laboratory, Great Britain)*, 1981.

[4] P. R. Lowrie. SYDNEY CO-ORDINATED ADAPTIVE TRAFFIC SYSTEM - PRINCIPLES, METHODOLOGY, ALGORITHMS. In *IEE Conference Publication*, number 207, pages 67–70. IEE, 1982.

[5] Suvrajeet Sen and K. Larry Head. Controlled optimization of phases at an intersection. *Transportation Science*, 31(1):5–17, 1997.

[6] Kok-Lim Alvin Yau, Junaid Qadir, Hooi Ling Khoo, Mee Hong Ling, and Peter Komisarczuk. A Survey on Reinforcement Learning Models and Algorithms for Traffic Signal Control. *ACM Comput. Surv.*, 50(3), jun 2017.

[7] Mohamed B. Trabia, Mohamed S. Kaseko, and Murali Ande. A two-stage fuzzy logic controller for traffic signals. *Transportation Research Part C: Emerging Technologies*, 7(6):353–367, 1999.

[8] K Aboudolas, M Papageorgiou, A Kouvelas, and E Kosmatopoulos. A rolling-horizon quadratic-programming approach to the signal control problem in large-scale congested urban road networks. *Transportation Research Part C: Emerging Technologies*, 18(5):680–694, 2010.

[9] Goof Sterk Van De Weg, Hai L. Vu, Andreas Hegyi, and Serge Paul Hoogendoorn. A hierarchical control framework for coordination of intersection signal timings in all traffic regimes. *IEEE Transactions on Intelligent Transportation Systems*, 20(5):1815–1827, 2019.

[10] Israel Alvarez Villalobos, Alexander S Poznyak, and Alejandro Malo Tamayo. Urban Traffic Control Problem: a Game Theory Approach. *IFAC Proceedings Volumes*, 41(2):7154–7159, 2008.

[11] Julio B Clempner and Alexander S Poznyak. Modeling the multi-traffic signal-control synchronization: A Markov chains game theory approach. *Engineering Applications of Artificial Intelligence*, 43:147–156, 2015.

[12] K Sekiyama, J Nakanishi, I Takagawa, T Higashi, and T Fukuda. Self-organizing control of urban traffic signal network. In *2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat.No.01CH37236)*, volume 4, pages 2481–2486 vol.4, 2001.

[13] I Nishikawa. Dynamics of oscillator network and its application to offset control of traffic signals. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, volume 2, pages 1273–1277 vol.2, 2004.

[14] Stefan Lammer, Hiroshi Kori, Karsten Peters, and Dirk Helbing. Decentralised control of material or traffic flows in networks using phase-synchronisation. *Physica A*, 363:39–47, 2006.

[15] Reik Donner. Emergence of Synchronization in Transportation Networks with Biologically Inspired Decentralized. In *Recent Adv. in Nonlinear Dynamics and Synchr.*, pages 237–275. 2009.

[16] Yoshiki Kuramoto. Self-entrainment of a population of coupled non-linear oscillators. In Huzihiro Araki, editor, *International Symposium on Mathematical Problems in Theoretical Physics*, pages 420–422, Berlin, Heidelberg, 1975. Springer Berlin Heidelberg.

[17] Monireh Abdoos, Nasser Mozayani, and Ana L.C. C Bazzan. Holonic multi-agent system for traffic signals control. *Engineering Applications of Artificial Intelligence*, 26(5-6):1575–1587, 2013.

[18] Sindhura Mandava, Kanok Boriboonsomsin, Matthew Barth, and Senior Member. Arterial Velocity Planning based on Traffic Signal Information under Light Traffic Conditions. *Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems on Intelligent Transportation Systems*, pages 160–165, 2009.

[19] Baisravan HomChaudhuri, Ardalan Vahidi, and Pierluigi Pisu. A fuel economic model predictive control strategy for a group of connected vehicles in urban roads. *American Control Conference*, pages 2741–2746, 2015.

[20] Xiaozheng He, Henry X Liu, and Xiaobo Liu. Optimal vehicle speed trajectory on a signalized arterial with consideration of queue. *Transportation Research Part C*, 61:106–120, 2015.

[21] Nianfeng Wan, Ardalan Vahidi, and Andre Luckow. Optimal speed advisory for connected vehicles in arterial roads and the impact on mixed traffic q. *Transportation Research Part C*, 69:548–563, 2016.

[22] Zhiyuan Du, Baisravan HomChaudhuri, and Pierluigi Pisu. Coordination strategy for vehicles passing multiple signalized intersections: A connected vehicle penetration rate study. *2017 American Control Conference (ACC)*, pages 4952–4957, 2017.

[23] Jackeline Rios-Torres and Andreas A. Malikopoulos. A Survey on the Coordination of Connected and Automated Vehicles at Intersections and Merging at Highway On-Ramps. *IEEE Transactions on Intelligent Transportation Systems*, PP(99):1–12, 2016.

[24] Lei Chen and Cristofer Englund. Cooperative Intersection Management: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):570–586, 2016.

[25] Elnaz Namazi, Jingyue Li, and Chaoru Lu. Intelligent Intersection Management Systems Considering Autonomous Vehicles: A Systematic Literature Review. *IEEE Access*, 7:91946–91965, 2019.

[26] Kurt Dresner and Peter Stone. A Multiagent Approach to Autonomous Intersection Management. *Journal of Artificial Intelligence Research*, 31:591–653, 2008.

[27] Seyed Alireza Fayazi and Ardalan Vahidi. Mixed-Integer Linear Programming for Optimal Scheduling of Autonomous Vehicle Intersection Crossing. *IEEE Transactions on Intelligent Vehicles*, 3(3):287–299, 2018.

[28] Pavankumar Tallapragada, Jorge Cortés, and Jorge Cortes. Hierarchical-Distributed Optimized Coordination of Intersection Traffic. *IEEE Transactions on Intelligent Transportation Systems*, PP:1–15, 2016.

[29] Chairit Wuthishuwong and Ansgar Traechtler. Consensus-based local information coordination for the networked control of the autonomous intersection management. *Complex & Intelligent Systems*, 3(1):17–32, 2017.

[30] Jackeline Rios-torres and Andreas A Malikopoulos. Automated and Cooperative Vehicle Merging at Highway On-Ramps. *IEEE Transactions on Intelligent Transportation Systems*, 18(4):780–789, 2017.

[31] Andreas A Malikopoulos, Christos G Cassandras, and Yue J Zhang. A decentralized energy-optimal control framework for connected automated vehicles at signal-free intersections. *Automatica*, 93:244–256, 2018.

[32] A M Ishtiaque Mahbub, L Zhao, D Assanis, and A A Malikopoulos. Energy-Optimal Coordination of Connected and Automated Vehicles at Multiple Intersections. In *2019 American Control Conference (ACC)*, pages 2664–2669, 2019.

[33] Alexander Katriniok, Peter Kleibaum, and Martina Joševski. Distributed Model Predictive Control for Intersection Automation Using a Parallelized Optimization Approach. *IFAC-PapersOnLine*, 50(1):5940–5946, 2017.

[34] F Molinari, A M Dethof, and J Raisch. Traffic Automation in Urban Road Networks Using Consensus-based Auction Algorithms For Road Intersections. In *2019 18th European Control Conference (ECC)*, pages 3008–3015, 2019.

[35] Fabio Molinari and Jörg Raisch. Automation of Road Intersections Using Consensus-based Auction Algorithms. *Proceedings of the American Control Conference*, 2018-June:5994–6001, 2018.

[36] Laleh Makarem and Denis Gillet. Model predictive coordination of autonomous vehicles crossing intersections. In *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2013.

[37] Alessandro Colombo and Domitilla Del Vecchio. Efficient Algorithms for Collision Avoidance at Intersections. In *Proceedings of the 15th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '12, pages 145–154, New York, NY, USA, 2012. Association for Computing Machinery.

[38] M R Hafner, D Cunningham, L Caminiti, and D Del Vecchio. Cooperative Collision Avoidance at Intersections:Algorithms and Experiments. *IEEE Transactions on Intelligent Transportation Systems (to appear)*, 14(3):1162–1175, 2013.

[39] Alessandro Colombo and Domitilla Del Vecchio. Least Restrictive Supervisors for Intersection Collision Avoidance: A Scheduling Approach. *IEEE Transactions on Automatic Control*, 60(6):1515–1527, jun 2015.

[40] Alejandro Ivan Morales Medina, Nathan Van De Wouw, and Henk Nijmeijer. Cooperative Intersection Control Based on Virtual Platooning. *IEEE Transactions on Intelligent Transportation Systems*, 19(6):1727–1740, jun 2018.

[41] Marco Di Vaio, Paolo Falcone, Robert Hult, Alberto Petrillo, Alessandro Salvi, and Stefania Santini. Design and Experimental Validation of a Distributed Interaction Protocol for Connected Autonomous Vehicles at a Road Intersection. *IEEE Transactions on Vehicular Technology*, 68(10):9451–9465, 2019.

[42] Ziran Wang, Guoyuan Wu, and Matthew Barth. Distributed Consensus-Based Cooperative Highway On-Ramp Merging Using V2X Communications. In *WCX World Congress Experience*. SAE International, apr 2018.

[43] Jakob Erdmann. Combining adaptive junction control with simultaneous Green-Light-Optimal-Speed-Advisory - IEEE Conference Publication. *2013 IEEE 5th International Symposium on Wireless Vehicular Communications (WiVeC)*, pages 1–5, 2013.

[44] Zhuofei Li, Lily Elefteriadou, and Sanjay Ranka. Signal control optimization for automated vehicles at isolated signalized intersections. *Transportation Research Part C: Emerging Technologies*, 49:1–18, 2014.

[45] Chunhui Yu, Yiheng Feng, Henry X. Liu, Wanjing Ma, and Xiaoguang Yang. Integrated optimization of traffic signals and vehicle trajectories at isolated urban intersections. *Transportation Research Part B: Methodological*, 112:89–112, 2018.

[46] Mehrdad Tajalli, Mehrzad Mehrabipour, and Ali Hajbabaie. Network-Level Coordinated Speed Optimization and Traffic Light Control for Connected and Automated Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–12, 2020.

[47] Biao Xu, Xuegang Jeff Ban, Yougang Bian, Wan Li, Jianqiang Wang, Shengbo Eben Li, and Keqiang Li. Cooperative Method of Traffic Signal Optimization and Speed Control of Connected Vehicles at Isolated Intersections. *IEEE Transactions on Intelligent Transportation Systems*, 20(4):1390–1403, 2019.

[48] Yi Guo, Jiaqi Ma, Chenfeng Xiong, Xiaopeng Li, Fang Zhou, and Wei Hao. Joint optimization of vehicle trajectories and intersection controllers with connected automated vehicles: Combined dynamic programming and shooting heuristic approach. *Transportation Research Part C: Emerging Technologies*, 98(March 2018):54–72, 2019.

[49] Qiangqiang Guo, Li Li, and Xuegang (Jeff) Ban. Urban traffic signal control with connected and automated vehicles: A survey. *Transportation Research Part C: Emerging Technologies*, 101:313–334, 2019.

[50] Manuel Rodriguez and Hosam Fathy. Speed Trajectory Optimization for a Heavy-Duty Truck Traversing Multiple Signalized Intersections: A Dynamic Programming Study. In *2018 IEEE Conference on Control Technology and Applications (CCTA)*, pages 1454–1459. IEEE, aug 2018.

[51] Behrang Asadi and Ardalan Vahidi. Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time. *IEEE Transactions on Control Systems Technology*, 19(3):707–714, 2011.

[52] P. G. Gipps. A behavioural car-following model for computer simulation. *Transportation Research Part B*, 15(2):105–111, 1981.

[53] Manuel Rodriguez and Hosam Fathy. Self-Synchronization of Connected Vehicles in Traffic Networks : What Happens When We Think of Vehicles as Waves ? *2019 American Control Conference (ACC)*, pages 2651–2657, 2019.

[54] Manuel Rodriguez and Hosam K Fathy. Distributed Kuramoto Self-Synchronization of Vehicle Speed Trajectories in Traffic Networks. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–11, 2021.

[55] Steven H Strogatz. From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators. *Physica D: Nonlinear Phenomena*, 143(1):1–20, 2000.

[56] Faraz Ashtiani, S. Alireza Fayazi, and Ardalan Vahidi. Multi-Intersection Traffic Management for Autonomous Vehicles via Distributed Mixed Integer Linear Programming. In *2018 Annual American Control Conference (ACC)*, pages 6341–6346. IEEE, jun 2018.

[57] Ali Jadbabaie, Nader Motee, and Mauricio Barahona. On the Stability of the Kuramoto Model of Coupled Nonlinear Oscillators †. pages 1–8, 2004.

[58] Liuhui Zhao and Andreas A. Malikopoulos. Decentralized Optimal Control of Connected and Automated Vehicles in a Corridor. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2018-Novem:1252–1257, 2018.

[59] Manuel Rodriguez, Xiangxue Zhao, Hayley Song, Anastasia Mavrommati, Roberto G Valenti, Akshay Rajhans, Pieter J Mosterman, Yancy Diaz-Mercado, and Hosam Fathy. A Gradient-Based Approach for Coordinating Smart Vehicles and Traffic Lights at Intersections. *IEEE Control Systems Letters*, 5(6):2144–2149, 2021.

[60] Mehran Mesbahi and Magnus Egerstedt. *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[61] Meng Ji and Magnus Egerstedt. Distributed coordination control of multiagent systems while preserving connectedness. *IEEE Transactions on Robotics*, 23(4):693–703, 2007.

[62] A M Ishtiaque Mahbub and Andreas A. Malikopoulos. Conditions to Provable System-Wide Optimal Coordination of Connected and Automated Vehicles. 2020.

[63] Steven G Shelby. Single-intersection evaluation of real-time adaptive traffic signal control algorithms. *Transportation Research Record*, 1867(1):183–192, 2004.

[64] Manuel Rodriguez and Hosam Fathy. Vehicle and Traffic Light Control Through Gradient-Based Coordination and Control Barrier Function Safety Regulation [under review]. *Journal of Dynamic Systems, Measurement and Control*, 2021.

[65] Wei Xiao, Calin Belta, and Christos G Cassandras. Decentralized Merging Control in Traffic Networks: A Control Barrier Function Approach. In *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*, ICCPS '19, pages 270–279, New York, NY, USA, 2019. Association for Computing Machinery.

[66] W Xiao, C G Cassandras, and C Belta. Decentralized Merging Control in Traffic Networks with Noisy Vehicle Dynamics: a Joint Optimal Control and Barrier Function Approach. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3162–3167, 2019.

[67] A D Ames, S Coogan, M Egerstedt, G Notomista, K Sreenath, and P Tabuada. Control Barrier Functions: Theory and Applications. In *2019 18th European Control Conference (ECC)*, pages 3420–3431, 2019.

[68] A D Ames, X Xu, J W Grizzle, and P Tabuada. Control Barrier Function Based Quadratic Programs for Safety Critical Systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2017.