Driverless Car and Multiagent Model Predictive Control

Csaba Bali

A dissertation submitted to the University of Bristol in accordance with the requirements for award of the degree of Doctor of Philosophy in the Faculty of Engineering

Department of Aerospace Engineering, CAME School

September 2019

Word count: Approximately fifty-one thousand words

Abstract

The field of robotics has made autonomous vehicles a reality. Their wide-scale deployment is expected to revolutionize transportation as we know it by improving traffic efficiency, reducing the number of road accidents, and lowering transportation-related costs. Moreover, it will provide social groups that are currently unable to drive independently with the opportunity to experience the benefits of personal transportation.

This work focuses on vehicle control at simple junctions in urban settings, challenging the limits of the optimal control technique of mixed-integer model predictive control. The challenging factor is the tendency for an exponentially growing number of potential discrete combinatorial choices to be considered as the number of discrete decisions (degree of freedom) in a problem increases. This imposes practical limitations on the number of vehicles, the length and resolution of future predictions, and the potential control configurations.

Vehicle junction crossing orders are incorporated into the problem, in order to find the optimal crossing order with respect to vehicle dynamics, constraints, and relative priorities. Formulations are shown for merging at Y junctions, crossing at cross junctions, and box junctions to remove deadlock situations. Control policies are shown starting with globally optimal model predictive control, preserving safe vehicle interactions with intuitive, simple time-headway safety constraints providing a recursive feasible control technique. For comparison, heuristic first-come-first-served and soft pre-merging policies are also developed.

Finally, simplifications of the mixed-integer formulations are shown for cross junctions to increase computational performance by exploiting the structure of the problem. The framework is further improved for future applications through added binary constraints and decentralised modification.

Acknowledgements

I would like to thank my supervisors, Professor Arthur G. Richards and Professor Robert J. Piechocki, for their time, support, and guidance as well as the examiners for their time, commitment, and valuable comments on my work.

My special thanks go to FARSCOPE CDT for its ambitious vision and the funding of EPSRC, which made this work possible.

I thank everyone whom I have had the luck and opportunity to meet, as they have all shaped my journey, my understanding, my life, and my PhD work. Thank you for helping in so many ways and for providing me with invaluable experiences and fond memories.

Finally, I would like to thank my family for their constant, never-ending support and belief.

Author's declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

Signed: Date:

Contents

List of Figures

1 Introduction

1.1 Motivation behind autonomous driving

Autonomous driving has garnered increasing attention from industry and the academic community. One of the first forms of driverless transportation was the horse, capable of easily navigating home with or without a rider. It remains a challenge to reliably reach this level of autonomy with today's vehicles given the fast-paced and complex nature of modern traffic. Advancements in computational performance (from software and hardware), miniaturisation, and robotics will soon bring society to a time when driverless vehicles are as common as motorised vehicles today. The predicted economic gains and social benefits from driverless technology are substantial, and it is a promising area for improvement given the staggering number of vehicles on the road. Recent data shows that, as a result of traffic congestion, the average driver annually loses 178 hours, costing them £1,317; that adds up to an annual national loss of about £8 billion [37]. Junctions act as bottlenecks in traffic flow, making them prime targets for improvements. Furthermore, about 38% of all fatalities stemming from road accident in the EU occur in urban areas; 20% of fatalities are caused by accidents in junctions [25]. According to [14], similar statistics are reported in the US. This is largely due to the universal bottleneck nature of intersections, which reduces traffic flows, increases the number of vehicle interactions, and, in turn, poses more difficult decision-making problems and requires greater driver attention. Autonomous vehicles have the potential to reduce driver-induced accidents, which according to the National Motor Vehicle Crash Causation Survey [73], constitute to 94% of all vehicle accidents in the US. Human-caused accidents have mainly been attributed to critical errors in recognition (41%), decision-making (33%), performance (11%), and non-performance (e.g. falling asleep) (7%) [73].

Safer traffic practices would reduce the rate of human injuries, material damages, and traffic delays, all of which currently come at a considerable social and economic cost. Human errors are expected to become far less prominent through the use of autonomous

vehicles, though current autonomous technology still requires some attention from drivers, making driving a shared responsibility. It is likely that a limited number of accidents will continue to occur, until fully independent, autonomous driving is achieved (e.g. Tesla Autopilot technology [7] and Uber driverless technology [80]).

Among the many benefits of autonomous vehicles, they will likely provide transportation options to those in certain social groups who currently do not have access to a car or unable to drive, such as young people, seniors, and disabled people.

Safety is paramount in the development of new algorithms—all other benefits of this technology are secondary objectives. This work aims to keep, for this reason, safety constraints hard in the control problem—they are not allowed to be violated. Other desired parameters or objectives are soft and optimised to match them as closely as possible (e.g. vehicle speed, comfort, cooperation).

Autonomous vehicle control at junctions, which entails challenging vehicle interactions, decision-making, and safety considerations has been chosen as the main topic for this work. The proposed techniques signify a potentially high impact, since they apply to the bottlenecks in traffic, junctions.

1.2 Overview of control for autonomous vehicles

Autonomous-vehicle technology was put to the test in early challenges sponsored by the United States' Defense Advanced Research Projects Agency (DARPA) as a way to promote research and innovation in state-of-the-art vehicle-control solutions [74]. While the initial challenges were based in the desert (off-road), later ones took place in urban settings [12, 46, 78]. Urban environments pose several unique and difficult problems for autonomous vehicles, such as uncontrolled junctions without traffic lights or signals like merging at Y junctions, crossing cross junctions, and box junctions. Numerous international projects on cooperative vehicle control has been collected by [14] for signalised and non-signalised intersections. Vehicle order, safe interaction, junction capacity, fairness, and deadlock-freeness are all considerations that must be made in the development of vehicle control. Each interaction between vehicles has some restriction on their joint state space (e.g. car-following, junction passing); these interactions can be viewed as obstacles that must be avoided in the relevant configuration space [45]. Therefore, vehicle motion and trajectory must be planned with care to avoid all forbidden vehicle states of interactions and obstacles. Numerous motion-planning approaches exist to calculate vehicle-motion plans, which are introduced well in [45]; specialised motion-planning methods for vehicle control in urban areas are collected in [54]. Some of the major areas in motion planning are covered by planning as a single-task, sampled techniques and trajectory planning. In [84], the control technique is based on Model Predictive Control (MPC) to achieve vehicle manoeuvres close to the physical limitations of the vehicle for high-speed collision

avoidance via swerving.

Trajectory planning requires solving two sub-problems, first a path-planning problem then a velocity-planning problem to gain the spatial then temporal elements of the trajectory. The decomposition of trajectory planning can help reduce the dimensionality of the original problem through the decoupling effect of sub-problems. The first sub-problem involves identifying the curvature-compliant path (that can be travelled by the vehicle); the second sub-problem involves obtaining the velocity profile over the path from the first sub-problem. In the context of a racetrack, vehicles are at their handling limit, meaning they must account for detailed upper-bound features of speed; in [15, 79] this is tackled using receding horizon control (MPC). In this work, the focus is on slow-speed urban settings, while the output of the first sub-problem determining the path is considered known; in general, the path can be calculated by a route planner, as the geometric design and layout of the roads can be obtained from public databases, such as the open-source OpenStreetMap [33]; however, in simple junction examples this is not necessary. The second sub-problem, velocity-profile optimisation, considers the vehicle dynamics and primarily deals with temporal constraints, for example, the moving car which only temporarily acts as obstruction.

The junctions addressed in this work can be represented by fixed state-space obstacles of interacting vehicles. A geometric-based branch-and-bound approach is shown for excluding fixed obstacles from the plan in [24] and finding optimal trajectories.

An early work considering linear optimal control of a merging junction, [4], considers all potential vehicle-order combinations. This present work uses model predictive control to obtain the optimal velocity profile while taking into account operational limits and dynamics as well as other merging constraints in Y junctions [6].

Today, the value of model predictive control techniques are generally recognised by industry for advanced control applications because they allow trajectories to operate more closely to the operational limits, resulting in a better performance [48]. Model predictive control inherently handles constraint formulations that are typical for any real application, as operational limits on certain process states and control inputs relate to finite limits on measures and actuation. The basics and theory of model predictive control are clearly and effectively discussed by [11, 59].

This thesis aims to create a safe and optimal multi-agent vehicle-control framework for uncontrolled junctions in low-speed urban traffic. Thanks to advancements in computer science and the development of more efficient, highly tailored algorithms, the optimisation problems can now be solved faster than ever and handle multiple agents with hybrid MPC, including difficult combinatorial decisions around junctions due to the discrete 'if-then' nature of many traffic rules. These discrete decisions are translated to Mixed-Integer-Program (MIP) optimisations. This enables formulations to incorporate discrete decisions in optimisation phase instead of using previously fixed integer decisions via approximation

or heuristics. The key elements discussed in this work are safety constraints, objective functions, multi-agent simulations, and performance and scaling considerations. Examples are based primarily around the simple atomic actions at uncontrolled junctions, such as merging at Y junctions, and crossing cross and box junctions commonly found in urban areas.

This work incorporates the order of vehicles merging or passing through an intersection into an optimisation through mixed-integer programming and big-M relaxation. For control purposes, [61] shows Mixed-Integer Linear Programming (MILP) with MPC to control a robotic agent.

Similar hybrid MPC formulations can consider discontinuous actuations, such as the throttle and brake actions in vehicle control [47].

Optimisation- and MPC-related works appear in junction vehicle control schemes [51]. The survey in [65] introduces related works involving intersections and merging junctions.

Alternatively, junction-control works can be based on time-space reservation algorithms. Dresner and Stone [22] detail an early reservation-based intersection-coordination policy in which the vehicles attempt to reserve grid tiles in the junction, which only one vehicle can occupy at any given time; the intersection node decides if the reservation is accepted or rejected based on simulations. [20] demonstrates a reservation-based algorithm in which only the first vehicle can request a reservation and arrival time and must cross the junction at full speed.

In [85], the authors develop an optimisation-based decentralised framework with a First-Come-First-Served (FCFS) policy. They implement approximate entry- and exit-time separation for vehicles in the junction area and show a trade-off between the fuel consumption and congestion level of two connected intersections. Similarly, [64] shows an optimisation-based framework with a FCFS policy and the same occupancy separation constraints for merging on highway on-ramps.

Threat-assessment techniques are surveyed in [17] for collision avoidance. Kamal et al. [39] detail a risk-function-based MPC framework that minimises speed errors and accelerations to avoid the risk of cross collision during intersection coordination.

A robust MPC scheme was proposed by [13] with a backup safety mode to abort the mission in a safe way in case of conflict; however, another controller is needed to restart the vehicle from the safety mode. Rizaldi and Althoff [66] list safety-rule considerations for vehicles to make them accountable for their road cooperation affecting actions. Junction safety in [43, 44] is shown with MPC formulations, where an infinite horizon contingency plan exists for the vehicles to maintain safety; additionally multiple vehicle ordering policies are detailed: rule based, FCFS, and concurrent crossing to fix the crossing orders.

Some closely related works employ mixed-integer formulations similarly to this thesis. Qian et al. [58] show a relative-priority-based MPC approach utilising brake-safe sets to

keep vehicles at safe distances from one another once their junction crossing order (relative priority) has been heuristically determined and fixed. This relative-priority based framework builds on the work [31] with results on acyclic priority graphs and trajectory planning around junction obstacles of vehicle pairs. Altche et al. [2] uses MILP to evaluate similar junction-related obstacles in 2D configuration spaces seeking minimal-time control of vehicles passing through junction areas. In another highly relevant work, Altche et al. [3] detail a state-of-the-art branch-and-bound optimisation of Mixed-Integer Quadratic Program (MIQP) implemented for semi-autonomous driving for the first time. Their work applies cooperative supervised vehicle control in junctions where the junction crossings orders (priorities) are not chosen heuristically but incorporated in the MIQP optimisation with integer-programming tools. These relevant methods of the above authors are available in more detail in thesis works [1, 57]. A two-stage MPC approach in [5] shows a non-convex operating-region optimisation using simulated annealing to select convex obstacle-free regions, that later, in the second stage, used by a convex fast MPC for safe robotic obstacle avoidance. This two-stage optimisation method was examined for the vehicle-ordering problem being discussed in this paper, but was found to be limited in its ability to handle a high number of vehicle-interaction hyperplanes.

In comparison, this thesis proposes a framework with a time-headway safety design, that has the collision sets inflated proportionally to the speed to achieve appropriate safety clearances during junction crossing. Furthermore, it incorporates integrated decision making for the vehicle ordering problem within the Mixed-Integer Model Predictive Control (MI-MPC), allowing the framework to obtain globally-optimal vehicle orders and control inputs at the same time instead of operating on some previously fixed orderings (from priority fixing approaches). Robustness of the control time delay and information propagation was addressed by forming spatio-temporal corner-cutting prevention constraints, as shown in [63]. Framework elements were tested with multiple cost formulations, policies, and additional junction-passing features; simplifications and added redundant constraints were included to increase computational performance. Finally, the control was reformulated in a decentralised manner for cases with restricted perception and information exchange.

1.3 Structure of the dissertation

Chapter 2 introduces the building blocks of the mixed-integer control framework. Starting with the route model and vehicle dynamics, time-headway safety considerations are introduced in the form of the positive control invariant set with respect to temporarily fixed road obstacles. The recursive feasibility of the MPC framework is shown to be closely related to vehicle safety and collision avoidance. The framework is designed with safety considerations to accommodate sudden stopping events of moving vehicles, achiev-

ing robustness against worst-case events, such as low-speed accidents. The design steps are demonstrated in an example of Y-junction merging, where the safety constraints are treated with the hyperplane formulation. Finally, the MI-MPC is formulated by employing big-M relaxation and separating hyperplane theorem. Soft priorities are shown in the decision-making process with relative weightings; examples of cooperation behaviours between vehicles are provided.

Chapter 3 discusses two formulations of the convex quadratic cost functions, showing a close connection between them. For intended operational cases, the two formulations are shown to result in equivalent optimisations and solution trajectories under given tuning conditions. The tuning of cost-weight parameters is designed with Linear Quadratic Regulator (LQR) theory in relation to the time-headway parameter yielding inherent stability results for single-agent cases. The characteristic that governs how vehicles slow down near obstacles is investigated in relation of time-headway parameter and horizon-length choices. An additional cost element is introduced for soft pre-avoidance to provide approximate early reactions to junction decisions (obstacles); its effect is shown on a pair of vehicles approaching a Y junction and merging. Finally, in the case of varying upper- and lower-bound speeds (e.g. around junctions, and at speed bumps), the MIP relaxation is shown with safe corner-cutting prevention.

In Chapter 4, the simulations are expanded to multi-agent vehicle-control cases for cross junctions. Examples are shown for a fixed number of vehicles looping within the junction simulations on double-O loops and 8-shaped loops. The potential for deadlock scenarios in these junctions due to their structural properties is discussed and displayed. A deadlock-free modification is then introduced in box junctions by extending the MIP framework. Various policies are then compared, those being a simple FCFS heuristic policy, the designed hybrid MPC, and the extended soft pre-avoidance early-reaction policy.

Chapter 5 examines performance-improvement methods for the cross-junction problem from Chapter 4. The problem can be changed to a simpler form, on account of the orthogonality between shared vehicle-interaction constraints, by removing the redundant hyperplanes. Two different approaches are explored in attempt to increase the speed of optimisation by adding redundant binary constraints. Finally, decentralised policies are formulated using sequentially shared future plans; trajectory densities accompany the given formulations for comparison.

Chapter 6 provides concluding remarks as well as potential future research directions that have emerged as a result of this thesis.

Supplementary material, result tables and notes are provided in the appendices.

1.4 Notations

This section provides a list of common notations that are used throughout this work alongside clarifying descriptions of respective definitions. Disambiguating comments are included where a single notation has multiple meanings and the context does not offer clear certainty. For example, notations in control theory and traffic analysis overlap in the case of Q which is used for both the quadratic cost matrix and traffic flow measure.

$\mathbb{Z}$	Integer numbers
$\mathbb{R}$	Real numbers
x	Vector, (e.g. $x \in \mathbb{R}^n$ real valued vector of size n, $x = [x_i,\ i = 1, \ldots, n]$)
x	Or concatenated decision vector, (e.g. $x = [x_n,\ n = 1, \ldots, N]$, where a state vector is x_n for the agent index n aggregated for compactness in multi-agent problems)
X	Matrix, ($X \in \mathbb{R}^{n \times m}$ real valued matrix of size $n \times m$)
$\square^{\mathrm{T}}$	Transpose, T in superscript
δt	Control period time, unit in seconds: [s]
t	Continuous time, unit in seconds: [s]
t_k	Discrete time at k index $t_k := k\delta t$, unit in seconds: [s]
$[x, y] = r(t)$	Position vector of a particle at time t in 2D Cartesian coordinates, unit in [m]
s	Arc length, one-dimensional position, measured along the path, unit in [m]
$[x, y] = \hat{r}(s)$	Map of arc length to 2D Cartesian coordinates, unit in [m]
J	Scalar cost (performance index)
$\square^*$	Optimal value (e.g. J^* optimal cost)
q, r, Q, R	Scalar and matrix weights in cost functions for quadratic states and control inputs
q_{f}, Q_{f}	Scalar and matrix weights of terminal states
Q, ρ	In traffic analysis, scalar measures of traffic flow $Q\ \left[\frac{\mathrm{veh}}{\mathrm{h}}\right]$ and density $\rho\ \left[\frac{\mathrm{veh}}{\mathrm{km}}\right]$
p, q or a, b	General vehicle indices commonly used in parameter or variable subscripts
$\{.,.\}$	Set of two or more elements (e.g. $\{p, q\}$ a set of two vehicle indices)
$(.,.)$	Ordered set of two or more elements (e.g. (p, q) two vehicles with car-following order)
N_{p}	Prediction horizon length
$\square_{\min}$, $\square_{\max}$	Minimum and maximum value of a parameter or variable, in subscript
$\mathcal{X}$	Allowed state set, $x \in \mathcal{X}$, commonly speed limits $\mathcal{X} = \{s, v \mid v_{\min} \leq v \leq v_{\max}\}$
$\mathcal{U}$	Allowed control set, $u \in \mathcal{U}$, commonly acceleration limits $\mathcal{U} = \{u \mid a_{\min} \leq u \leq a_{\max}\}$

1.5 Acronyms

MPC	Model Predictive Control
MIP	Mixed-Integer Program
MI-MPC	Mixed-Integer Model Predictive Control
MILP	Mixed-Integer Linear Program
MIQP	Mixed-Integer Quadratic Program
LTI	Linear Time Invariant
LQR	Linear Quadratic Regulator
FCFS	First-Come-First-Served

2

Vehicular control with time-headway MI-MPC

This chapter details the main steps and considerations for mathematical representation of the safe baseline control for automated vehicles at intersections. It serves to systematically introduce the building blocks, methodologies, and modelling approaches used for the proposed control system, which is further developed and analysed in later chapters. The work in this chapter was published in [6], though, extended and more detailed investigations are presented here. The main contribution of this chapter is the derivation of safe, control-invariant, simple time-headway parameters. Moreover, it shows the design of a mixed-integer optimal control framework using the control-invariant sets on a merging obstacle that appears in urban Y-shaped junctions.

Publication resulted from this chapter: [6].

2.1 Model predictive control

A standard nominal MPC optimisation is shown below:

MPC-2.1

$$J^{*}_{\text{MPC-2.1}} = F(x(k+N_{\text{p}}|k)) + \min \sum_{j=0}^{N_{\text{p}}-1} l\left(x\left(k+j|k\right), u\left(k+j|k\right)\right)$$

$$\begin{aligned}
\text{s.t.} \quad & \forall j \in \{0 \dots (N_{\text{p}}-1)\} : \\
& x\left(k|k\right) = x\left(k\right) \\
& x\left(k+j+1|k\right) = f(x\left(k+j|k\right), u\left(k+j|k\right)) \\
& x\left(k+j+1|k\right) \in \mathcal{X} \\
& u\left(k+j|k\right) \in \mathcal{U},
\end{aligned}$$

where the state and control inputs are x and u, respectively; the cost function, J, is minimised and consists of the summation of stage costs $l(x, u)$ and the terminal cost $F(x)$; control times are at $t_k = t(k) := k\delta t$, where the discrete time is $k \in \mathbb{Z}$ and the positive sampling or period time is δt; process dynamics are defined by the discretised model $x(k+1) = f(x(k), u(k))$ while $x \in \mathcal{X}$ and $u \in \mathcal{U}$, where the allowed state and control sets are $\mathcal{X}$ and $\mathcal{U}$, respectively.

2.2 Problem definition

The aim is to tackle vehicle-control problems for a set of $\mathcal{N} = \{1, 2, \ldots, N\}$ digitally controlled vehicles (e.g. automated or autonomous cars). Urban traffic environments were selected where the fixed road-network layout is known and vehicle interactions happen at relatively slow speeds (< 10 ms^{-1}), with focus on uncontrolled and non-signalised junctions. Low traffic speeds provide more time for decision-making in the control optimisation while the relatively high vehicle density on roads gives way to more vehicle interactions and non-trivial, intricate situations. The simple safety approach developed in this work is based on time-headway separation; this separation approach is valid primarily for low-speed environments; it becomes inefficient in high-speed traffic due to longer than necessary inter-vehicle separation gaps. Other model simplifications serve to neglect quadratic speed-dependent terms in vehicle dynamics, such as air drag and lateral dynamics on curved paths (assuming curvature-compliant, reasonably planned paths). Specific characteristic dynamics may appear when driving slower than a rolling stop, such as probabilistic stopping, which is a sudden-stop event at near zero speed. Unmodelled friction terms and tribological properties are responsible for probabilistic stopping which present through the drivetrain and during tire-pavement contact. Sudden-stop events, as with probabilistic stopping, occur when a vehicle experiences an unexpected state evolution and abruptly decelerates from rolling to a stationary state. In a worst-case scenario, this could be an accident or other non-operational emergency situations ahead of the controlled vehicle, necessitating a safe response. Nominal MPC with hard output constraints like obstacle or collision avoidance may easily become infeasible in the presence of sudden stop events or other model mismatch errors. The rate of model mismatch grows alongside the number of simplifications and assumptions common at higher levels of abstraction where also longer control period times are dominant; this is a practical rule for high level controllers in the control hierarchy, usually considering slowly changing model dynamics over a longer period of time [69].

The applied MPC approach is designed as a mid- to high-level controller that tackles relatively slow (1–10 s) mission-specific trajectories. Lower, specialised controllers are developed to accept reference controls and trajectories from the MPC [69]; moreover, they serve to handle unmodelled, high-frequency dynamics, such as the engine, braking,

steering and suspension control. These are exempt from modelling due to their frequency, complexity and moderate influence on the modelled time scale.

For describing and controlling vehicles, vehicle motion is calculated in a simplified approach for a single particle representing a vehicle as a point-mass. The function of position coordinates (position vector) $r(t)$ is to describe the 2D spatial position vector of the vehicle particle on the Cartesian plane at a given time t (or, if terrain is involved, in 3D space); thus, the motion of the particle is considered known:

$$\begin{bmatrix} x \\ y \end{bmatrix} = r(t) \in \mathbb{R}^2. \tag{2.2}$$

A possible decomposition of the motion into two sub-elements is based on the determination of its graph (the path) and the travel plan along this path. The path is a time-ordered set of coordinates for particle positions. It is worth noting that the path alone does not define the motion, as it is missing the temporal element of the motion.

Let the motion of the particle alternatively be parametrised with arc length s, where the arc length may be interpreted as travelled distance or a one-dimensional position measured along the path:

$$r(t) = \hat{r}(s(t)), \tag{2.3}$$

where the map is $\hat{r}$ and the scalar-valued travel plan is $s(t)$ with the temporal element of the motion. Consider that the $\hat{r}$ map is readily available in most urban road networks and generated by a route planner, where vehicle path and route elements consist of simple road geometries yielding fixed paths to the mission goal. Figure 2.1 shows how vehicle n traverses a road network; its position coordinates are described uniquely by the arc-length (one-dimensional position) parameter along its fixed route. One frequent approach to obtain $s(t)$ is to use speed, which is the derivative of $s(t)$ with respect to time:

$$v(t) = \frac{ds(t)}{dt}, \tag{2.4}$$

where speed is $v(t)$ and scalar with a ms^{-1} unit; furthermore, the speed is the absolute value of the velocity vector $\dfrac{dr(t)}{dt} \in \mathbb{R}^2$.

Trajectory in robotics refers to the fully defined particle motion consisting of both spatial and temporal elements. In mathematics, trajectory is an ordered set of general states mapped onto the same set of states.

Trajectory planning in robotics is concerned with finding the path and related velocity profile for robots [45]. The decomposition of this duality is shown in an early work, [40], where trajectory-planning is decomposed into two sub-problems. First, it involves solving a planning problem to obtain a path that avoids conflicts with static obstacles. Second, it

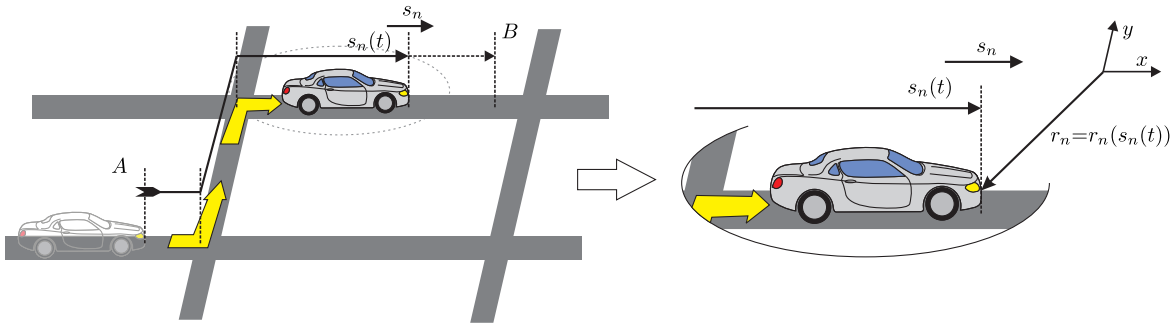


Figure 2.1: Vehicle route from A to B over the road network (with position definitions).

involves solving a velocity-planning problem, to describe the motion over the fixed path from the first step, to avoid conflicts with temporary, moving obstacles that could impose conflicting states at certain time intervals. In [15, 79], semi-analytical velocity-profile-optimisation problems are shown with MPC (receding horizon control) formulation along the fixed path of racetracks.

Fixed paths in urban environments are obtained using a mission or route planner. These paths are planned between the starting (current) position and the destination position (Figure 2.1). The path-planning that calculates the fixed path ($\hat{r}$ map) is not considered further in this work; the path is instead treated as a known, since examples making use of it are simplistic to the point of triviality (e.g. traversing on a one-way road). However, the path is assumed to be feasible in respect of vehicle kinematics, operational limits, and lateral accelerations (e.g. curvature compliant with respect to corner geometries and permanently static features, such as pavement shape or parked vehicles). When the path is not feasible, it is not guaranteed that the mission goal of the vehicle would be reached in a finite time without replanning the path.

$s(t)$ arc length parametrisation can be linearly transformed, without loss of generality (e.g. arbitrarily shifted and scaled to obtain a generalised position parameter). In this work, s is simply referred to as position, with the default unit in meters [m] and will be shifted to arbitrarily place the origin at the junction of interest. In multi-degree-of-freedom robotic systems, the number of general positions (N) describes application-specific measures, joint degrees, linear positions, etc., which lead to an intuitive N-dimensional coordination-space representation. Within this coordination space, interference between coordinates, such as robotic arm links, are represented as obstacles, as shown in [72].

The N-dimensional general position space in a traffic system describes agent positions for each considered vehicle. Additionally, 2D pairwise vehicle-coordination planes [45] can be used to represent unwanted vehicle-interference positions (e.g. collision of finite vehicle bodies); see [30, 31, 58] where the obstacles are convex approximations of the real conflicting positions. The configuration space for three vehicles is shown in Figure 2.2, with the grey obstacle bodies representing convexified collision sets. Alternatively, normalised space representation appears in the literature, where s is defined over $[0, 1]$ range, which

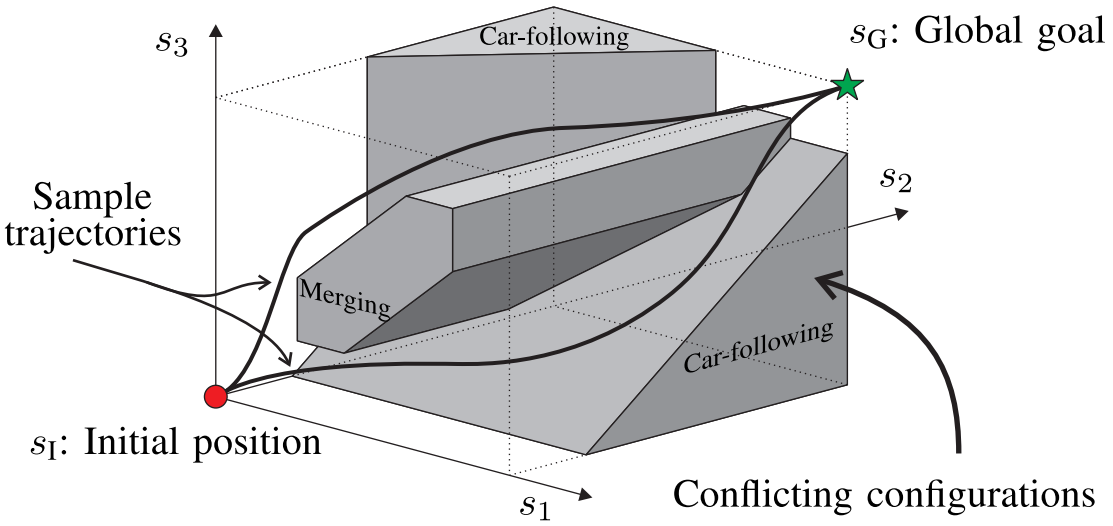


Figure 2.2: Representation of coordination space for three cars where each axis is a vehicle position, with the following collision configurations: vehicle 1 and 3 must first merge before separating; both are pursued by vehicle 2 and they must stay ahead of it.

provides a natural sense of the completed distance of a plan (e.g. in percentages through linear mapping). The solution to the overall problem would require a feasible position curve just like it is indicated in Figure 2.2 where either of the two possible sample trajectories would suffice. This curve is required to reach the s_G global goal position without intersecting the collision configuration (set) for all trajectory (vehicle-order) choices.

Thus, the existence of such a trajectory excluding all obstacles is assumed from the starting point s_I to the endpoint s_G. In cases with a high number of traffic participants using the road network (e.g. within a country) the dimensionality of this representation is staggering. However, geographically distant vehicles are not likely to directly interact, as their start and goal positions may be in different cities and their routes may never intersect. In order to maintain a reasonable scale for this problem, the space of interest is restricted to a local subset of vehicles—a junction and its vicinity. The selected vehicles are likely to interact due to close spatial and temporal proximity. The physical size of the region of interest could change depending on significant parameters, such as vehicle speeds and information availability. This region of interest and the scenarios within it are initially treated with a locally centralised control problem; for this reason, a limited number of agents are used to limit the number of interactions and maintain reasonably low computational complexity. Some previous works have considered junctions as supervision areas, such as [2], in which boundary conditions are specified at the start and end position of the simulated area with state constraints restricting flexibility. In this work, however, initial states are free choices so long as they respect safety criteria constraints introduced later in this chapter.

2.2.1 Vehicle dynamics

Longitudinal dynamics describe the state evolution of position variables and constitute a common simplification-based modelling approach; it is commonly used in transportation, such as in [52], and is even used at junctions with curved roads [19, 26]. Thus, the model is an ideal projection of motion to a spatial dimension (i.e. on the axis of s position). The governing differential equations are:

$$\dot{s}(t) = v(t), \tag{2.5}$$

$$\dot{v}(t) = a(t), \tag{2.6}$$

where the instantaneous speed is $v(t)$ along the path and defines the trajectory while the longitudinal (tangential) component of acceleration is $a(t)$, this is valid $\forall n \in \mathcal{N}$ agents. When evident, details on measures, variables and parameters are omitted from the notations to make the equations general and compact and keep the focus on significant relations (e.g. $s(t)$ signifies $s_n(t)$ an agent specific position in general without n subscript index).

Continuous state evolution of agents is expressed in the Linear Time Invariant (LTI) form:

$$\dot{x}(t) = A_{\text{con}}x(t) + B_{\text{con}}u(t), \tag{2.7}$$

where the continuous system and control matrices are $A_{\text{con}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ and $B_{\text{con}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and state vector $x(t) = [s(t),\ v(t)]^{\text{T}}$ while $u(t) = a(t)$. Through discretisation with δt discrete time using constant acceleration (Zero-Order-Hold (ZOH)), the following LTI system is obtained:

$$x^{k+1} = f(x(k), u(k)) := Ax^k + Bu^k \tag{2.8}$$

where the discrete system and control matrices are $A = \begin{bmatrix} 1 & \delta\text{t} \\ 0 & 1 \end{bmatrix}$ and $B = \begin{bmatrix} \frac{\delta\text{t}^2}{2} \\ \delta\text{t} \end{bmatrix}$, the vector of discrete-time states is $x^k = [s^k, v^k]^T$ (i.e. position and speed); the control input is $u^k = a^k$ with longitudinal acceleration a^k. An alternative, shorter notation of the discrete-time argument is indicated by superscript $\square^k$ (e.g. $u^k = u(t_k)$). At this point, each vehicle is assumed to have time-synchronous control. State limits are defined on speed:

$$0 \leq v \leq v_{\max}, \tag{2.9}$$

yielding that, for simplicity, vehicles must abide by some constant upper limit on speed

and are part of uni-directional traffic, meaning they cannot move backwards on their path. Limitations in lateral dynamics may drive variations in the upper bounds of speed, which is important for race cars on curvilinear tracks or minimal-time velocity optimisation, as in [15]. However, the scenarios being discussed are in low-speed urban environments, meaning there is no real need to drive at the physical limits of the vehicle—lateral dynamics constitute less of a dominant factor in these situations. The speed limits are respected in $t \in [t_k, t_{k+1}]$ by defining (2.9) at both the start and end of the single control period (i.e. t_k and t_{k+1}). This is evident, as the continuous-time speed function $v(t)$ is linear on $t \in [t_k, t_{k+1}]$ because the LTI (2.8) with constant acceleration (ZOH) and the integral relation from (2.6) result in a line segment. Thus, the full line segment remains within the speed limits, since the points on this line segment (inter-sample speeds) are the linear combination of its extrema. Moreover, simple dynamics constraints are assumed, such as:

$$F_{\max \mathrm{bra}} \leq F \leq F_{\max \mathrm{tra}}, \tag{2.10}$$

where the longitudinal force is F, which acts on the vehicle; the maximum braking force limit is $F_{\max \mathrm{bra}}$; the maximum tractor force limit is $F_{\max \mathrm{tra}}$. Through simplification with the non-changing mass of vehicle, the acceleration limits are:

$$a_{\min} \leq a \leq a_{\max}, \tag{2.11}$$

where the maximum deceleration is $a_{\min} < 0$ (lower limit on acceleration) and the maximum acceleration limit is $a_{\max} > 0$.

As a result of the deterministic vehicle model, new state predictions can be made with (2.8) simply as equality constraints; moreover, (2.9) and (2.11) provide inequality constraints for simple speed and acceleration limits. These are readily incorporated in $\mathcal{X}$ and $\mathcal{U}$ sets in MPC 2.1 optimisation as $\mathcal{X} = \{s, v \mid v_{\min} \leq v \leq v_{\max}\}$ and $\mathcal{U} = \{u \mid a_{\min} \leq u \leq a_{\max}\}$ or could be imposed as part of the linear matrix inequalities. The following section discusses the construction of safety sets and constraints for various vehicle interactions.

2.3 Obstacle handling

Stationary and static obstacles, such as road features and parked vehicles, are included in the fixed-path plan and are known in the problem. Thus, obstacle avoidance in the velocity optimisation is concerned with moving or mobile obstacles at certain locations at certain times, such as other vehicles on the roads and in junctions. As previously discussed, the complete trajectory plan is determined once the velocity optimisation is solved and feasible. Following the approximated obstacle formulation of [30], convex collision sets

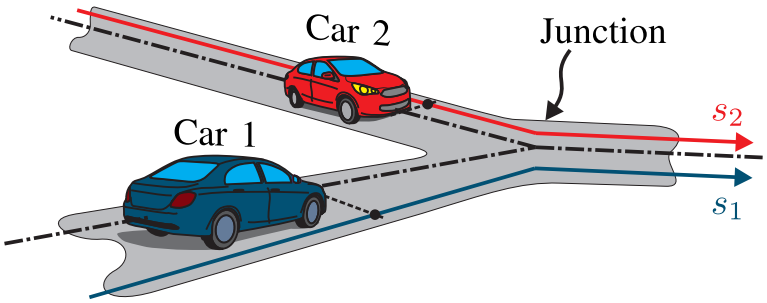


Figure 2.3: Schematic of a two-car merging scenario; s_1, s_2 are positions along route. [6]

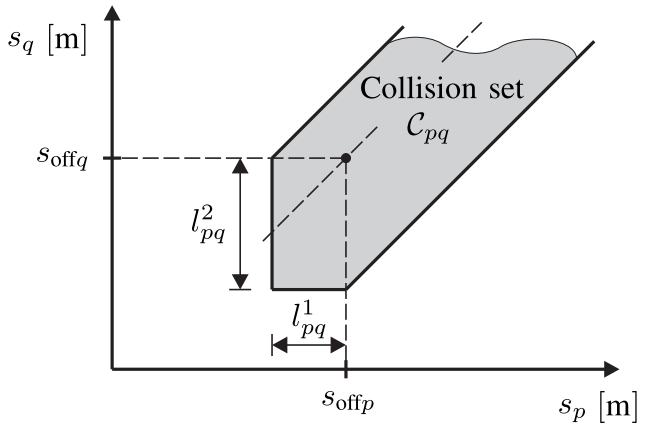


Figure 2.4: $\mathcal{C}_{pq}$ collision set indicated for two merging vehicles. [6]

are defined around the undesirable positions of each conflicting vehicle pair $\{p, q\} \in \mathcal{N}$ such that $p \neq q$. This set of positions includes those where the vehicle body frames would overlap, indicating physical contact and collision. Without a loss of generality, the interest is initially a single merging, as shown in Figure 2.3. Consequently, the convex bounding collision set of a merging interaction is a single joint polyhedron defined as:

$$\mathcal{C}_{pq} := \{x_p, x_q | s_p > L_1,\ s_q > L_2,\ s_p - s_q > L_3,\ s_q - s_p > L_4\}, \tag{2.12}$$

where constants L_i, $i = \{1, \dots 4\}$ determine the shape and position[1] of the merging obstacle over the s_p–s_q 2D configuration plane (see Figure 2.4).

Pairwise vehicle conflict is defined as:

$$\mathcal{C}_{pq} \neq \emptyset, \tag{2.13}$$

and the collision with this obstacle is defined as:

$$[x_p, x_q] \in \mathcal{C}_{pq}. \tag{2.14}$$

[1]For the merging obstacle without loss of generality, the constants incorporate the position offsets $L_1 := s_{\text{off}p} - l^1_{pq}$, $L_2 := s_{\text{off}q} - l^2_{pq}$, $L_3 := s_{\text{off}p} - s_{\text{off}q} - l^1_{pq}$, $L_4 := s_{\text{off}q} - s_{\text{off}p} - l^2_{pq}$ (shown in Figure 2.4).

Hence, the imposed constraint for the vehicle states are:

$$[x_p, x_q] \notin \mathcal{C}_{pq} \tag{2.15}$$

or, alternatively:

$$[x_p, x_q] \in \overline{\mathcal{C}_{pq}}, \tag{2.16}$$

where the complement of $\mathcal{C}$ is $\overline{\mathcal{C}}$.

Recursive feasibility and vehicle safety are strongly related to collision-free vehicle control where invariant set theory is employed to obtain theoretical guarantees.

2.3.1 Invariant sets

The recursive feasibility property is essential for safety-critical optimisation problems, in which guaranteed constraint satisfaction must be ensured at all costs [42]. The proposed framework considers safety constraints to ensure vehicle separation from obstacles and from other vehicles with (2.16).

Following the definition from [42] for a discrete time system $x^{k+1} = f(x^k, u^k)$:

$$\Omega \text{ is control invariant} \Leftrightarrow \forall x \in \Omega,\ \exists u \in \mathcal{U} \text{ such } f(x, u) \in \Omega. \tag{2.17}$$

For simplicity, the vehicle in consideration must remain in the set of safe states to be able to stop before a static obstacle position (e.g. the goal position, a junction entry, or behind another (temporarily) stationary vehicle).

2.3.2 Headway

As a preliminary to constructing the safe invariant set, spatial and temporal measures for the distance between vehicles are introduced.

The spatial distance, 'distance headway', is the distance between the corresponding reference points of two vehicles following each other at a given time assuming the same vehicular paths. Thus, $d_{\mathrm{h}}(t) = s_{\mathrm{l}}(t) - s_{\mathrm{f}}(t)$, where the longitudinal positions on the road for the leader vehicle is s_{l} and, for the follower vehicle, is s_{f}. The leader vehicle is, by definition, ahead of the follower; thus, $d_{\mathrm{h}} > 0$. The distance gap measure, or simply the gap, is the clearance between vehicles defined as $d_{\mathrm{g}}(t) = s_{\mathrm{l}}(t) - L_{\mathrm{l}} - s_{\mathrm{f}}(t)$, where the leader vehicle length is L_{l} and the vehicle reference points are at the front bumpers; the positive gap or clearance $d_{\mathrm{g}} > 0$ is a stricter constraint assuming finite length vehicle bodies.

The temporal distance, time headway, is the temporal counterpart of 'distance headway'; it is similarly used as a method of analysis for interpreting traffic-flow data [75]. Gross time headway, is measured between the corresponding points of vehicles reaching

the same position on the road (i.e. $t_{\mathrm{gh}} = t_{\mathrm{f}} - t_{\mathrm{l}}$, $s_{\mathrm{f}}(t_{\mathrm{f}}) = s_{\mathrm{l}}(t_{\mathrm{l}})$). Net time headway, or the time gap, is the time span between the front bumper of the follower vehicle reaching the position of the rear bumper of the leader vehicle (i.e. $t_{\mathrm{nh}} = t_{\mathrm{f}} - t_{\mathrm{l}}$, $s_{\mathrm{f}}(t_{\mathrm{f}}) = s_{\mathrm{l}}(t_{\mathrm{l}}) - L_{\mathrm{l}}$). As traffic-flow analysis is done on time-space diagrams of existing vehicle-trajectory data, the headway measures are relatively straightforward to read off of the graphs on the corresponding axes (i.e. t-time and s-space).

However, real-time control cannot readily ascertain the current gross or net time headway between vehicles without making additional assumptions to predict future vehicle behaviours and trajectories. This inherently causes a prediction error in the measure, as in general predictions may differ from the course of real events and states.

In this thesis, 'time headway' is used to refer to the similar concept of net time headway, or the time gap, between the controlled (follower) vehicle and the worst-case assumption of the leader's rear-bumper position interpreted as a stationary obstacle. This relates to the conservative approach of worst-case dynamics for instantaneous stops of the leader vehicles. It is a cautious approach, as the capabilities and actions of the leader vehicles are not necessarily known in advance by the follower vehicles. Thus, in this work, time headway t_{h} is a worst-case vehicle-specific parameter, rather than a data-analysis tool for post-processing. The t_{h} parameter will be extensively used in this work to ensure safe clearances and prevent collisions.

2.3.3 Simple time-headway invariant set

The positive control-invariant set (referred to as the invariant set for short) is defined using the intuitive time headway-formulation. In the case of highway car-following, this can be found in the UK Highway Code as the 'two-second rule' [77, Rule 126]. The time-headway (constant-time-gap) policy relates to an intuitive human driving style [27].

Theorem 1. *Let* $\Omega := \{s, v \in \mathbb{R}^2,\ 0 \leq v \leq v_{\max},\ s + t_h v \leq s_{obs}\}$ *for the discrete system with kinematics (2.8) and* $u \in [a_{\min}, 0]$, $a_{\min} < 0$ *if* $0 < \delta t \leq 2t_h$, $t_h \geq \frac{v_{\max}}{-a_{\min}} - \frac{\delta t}{2}$ *for* $t_h > 0$ *headway time and* $\delta t > 0$ *discrete time step then* Ω *set is positive control invariant for this discrete system. (Figure 2.5)*

Proof. Let the initial states of a vehicle $x^k \in \Omega$ set, where $x_{\mathrm{v1}} = [s_{\mathrm{obs}}, 0]^T$ and $x_{\mathrm{v2}} = [s_{\mathrm{obs}} - \mathrm{t_h} v_{\max}, v_{\max}]^T$ are the vertices of the convex polyhedral set. According to [9], it is a sufficient proof of invariance for a polyhedral set and a discrete LTI system when all polyhedral vertices, after a discrete-time evolution, satisfy all the constraints of that polyhedral set. Thus, for the proof of positive control invariance (2.17) with state evolution (2.8), the test cases of the vertices are:

Vertex v1: $x^k = x_{\mathrm{v1}} = [s_{\mathrm{obs}}, 0]^T$

$$s^{k+1} = s_{\mathrm{obs}} + \frac{1}{2}a^k \delta \mathrm{t}^2, \tag{2.18}$$

$$v^{k+1} = a^k \delta \mathrm{t}, \tag{2.19}$$

for the trivial solution of $a^k = 0$ the states remain unchanged, satisfying $x^{k+1} \in \Omega$.
Vertex v2: $x^k = x_{\mathrm{v2}} = [s_{\mathrm{obs}} - \mathrm{t_h} v_{\max}, v_{\max}]^T$

$$s^{k+1} = s_{\mathrm{obs}} - \mathrm{t_h} v_{\max} + v_{\max} \delta \mathrm{t} + \frac{1}{2}a^k \delta \mathrm{t}^2, \tag{2.20}$$

$$v^{k+1} = v_{\max} + a^k \delta \mathrm{t}. \tag{2.21}$$

The new states must satisfy each of the constraints defining Ω. Inequality $0 \leq v^{k+1}$ from (2.9) combined with (2.21) yields:

$$\delta \mathrm{t} \leq \frac{v_{\max}}{-a^k}, \quad \text{for} \quad a^k < 0. \tag{2.22}$$

The constraint of $v^{k+1} \leq v_{\max}$ from (2.9) with (2.21) is satisfied for $a^k \leq 0$ control choice. The final constraint gives:

$$s^{k+1} + \mathrm{t_h} v^{k+1} \leq s_{\mathrm{obs}}. \tag{2.23}$$

Using (2.20), (2.21) and (2.23):

$$v_{\max} + a^k \left(\frac{1}{2}\delta \mathrm{t} + \mathrm{t_h} \right) \leq 0 \tag{2.24}$$

by substituting the maximum available deceleration (minimum acceleration):

$$\frac{v_{\max}}{-a_{\min}} \leq \frac{1}{2}\delta \mathrm{t} + \mathrm{t_h}. \tag{2.25}$$

Moreover, by combining (2.22) and (2.24), they give:

$$v_{\max} + a^k \left(\frac{1}{2}\delta \mathrm{t} + \mathrm{t_h} \right) \leq 0 \leq v_{\max} + a^k \delta \mathrm{t}, \tag{2.26}$$

$$\delta \mathrm{t} \leq 2\mathrm{t_h}, \tag{2.27}$$

for $a^k < 0$. □

In Figure 2.6, $\mathrm{t_h}$–$\delta \mathrm{t}$ parameter regions are shown where Ω set is a positive control-invariant using (2.25) and (2.27).

Remark: Ω can be closed by another side from the left (assuming safe non-empty set (i.e. $s_c < s_{\mathrm{obs}}$)) at arbitrary $s_{\mathrm{c}} < s^k$, resulting in two more vertices ($x_{v3} = [s_{\mathrm{c}}, 0]^T$,

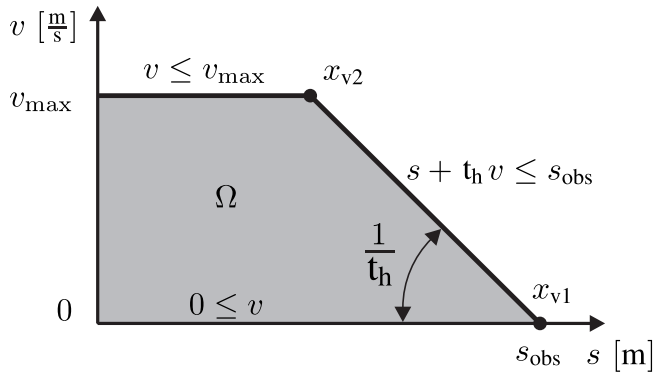


Figure 2.5: Shaded area represents the Ω set. [6]

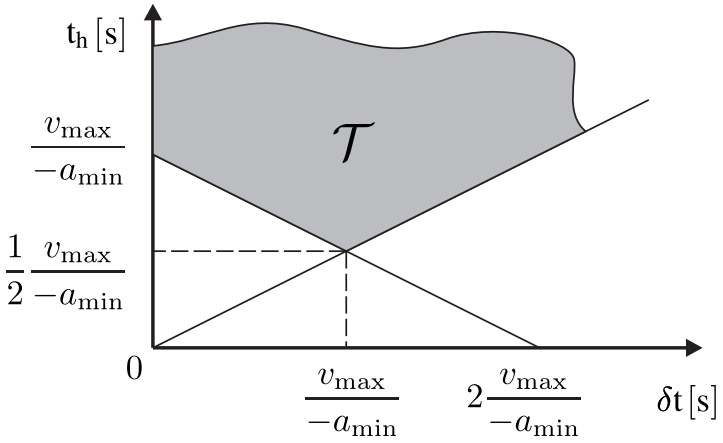


Figure 2.6: The shaded region indicates t_{h}–δt parameter choices that ensure Ω set to be a positive control-invariant set. [6]

$x_{v4} = [s_{\mathrm{c}}, v_{\max}]^T$) if $s_c < s_{\mathrm{obs}} - t_{\mathrm{h}} v_{\max}$ or, otherwise, three vertices overall. However, these instances are covered by the trivial solutions of previous case studies of vertices and by the monotonic rule of $s_{\mathrm{c}} < s^k \le s^{k+1} \le s_{\mathrm{obs}}$; thus, they can be disregarded from further problem formulation.

A comparison between the introduced Ω invariant set is shown in Figure 2.7 against the maximal invariant set with the same vehicle properties. Both Ω sets with $\delta t \approx 0$ and 0.5 s were chosen with the minimal allowed t_{h}. The maximal invariant set was computed iteratively using the dual of a one-step reachable set [11]:

$$\mathrm{Pre}(\mathcal{S}) \triangleq \{x \in \mathrm{R}^n,\ \exists u \in \mathcal{U} \text{ s.t. } f\,(x, u)\}\,, \tag{2.28}$$

where the initial $\mathcal{S}_0 = \left\{[0,\ 0]^{\mathrm{T}}\right\}$ is the single rightmost corner point (i.e. the closest stationary vehicle position). The maximal invariant set was iteratively calculated backwards with:

$$\mathcal{S}_{k-1} = \mathrm{Pre}(\mathcal{S}_k) \cap \mathcal{X}, \tag{2.29}$$

where the representation of the permitted speed range is incorporated in the $\mathcal{X}$ set.

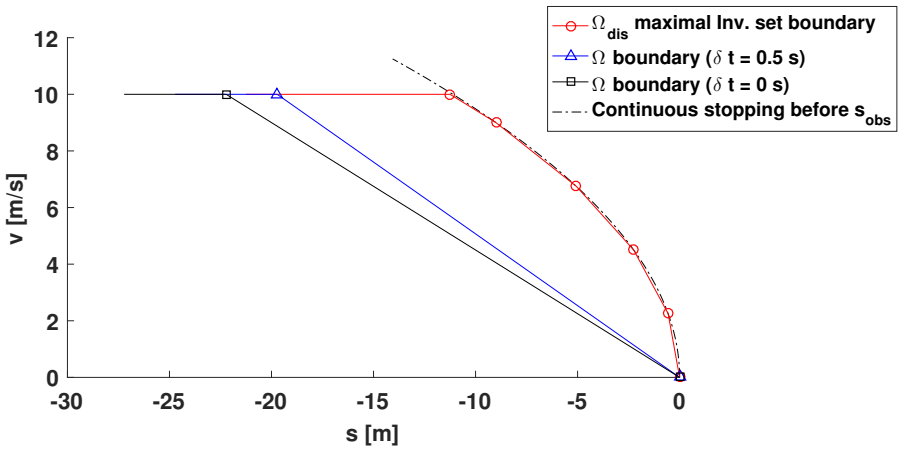


Figure 2.7: Comparison of discrete positive control-invariant sets generated with $v_{\max} = 10\ \mathrm{ms}^{-1}$, $a_{\min} = -4.5\ \mathrm{ms}^{-2}$, $s_{\mathrm{obs}} = 0$, $\delta t = 0.5\ \mathrm{s} \leq -v_{\max}/a_{\min}$ and $\delta t = 0\ \mathrm{s}$

Furthermore, Figure 2.7 illustrates the distance gap between the rightmost maximum speed, $v_{\max}$, of the Ω-invariant set and the continuous physical invariant stopping set increase for higher speeds. This gap is the reason why simple time-headway models are acceptable for low-speed traffic and why, conversely, they are overly conservative for high-speeds; high-speed models must also consider higher order terms and tuned parameters to properly describe the physical behaviours in wide range around the operating work-point: (i.e. desired speed).

2.3.3.1 Numerical example: Safe stop

In the following example, a vehicle approaches an obstacle at $s_{\mathrm{obs}} = 50$ m using an Ω invariant set while satisfying (2.9) and (2.11) operational limits. The control is only calculated for a single time step ahead ($N_p = 1$) in a reactive control fashion, maximising the speed.

MPC-2.2

$$
\begin{aligned}
&\min \ |v_{\max} - v\,(k+1|k)| \\
&\text{s.t. :} \\
&\qquad x\,(k|k) = x\,(k) \\
&\quad x\,(k+1|k) = A\,x\,(k|k) + B\,u\,(k|k) \\
&\quad x\,(k+1|k) \in \Omega \\
&\qquad\qquad x \in \mathcal{X} \\
&\qquad\qquad u \in \mathcal{U}
\end{aligned}
$$

Notice the simple l_1 norm objective function in control MPC 2.2, which has a connection to the terminal position objective.

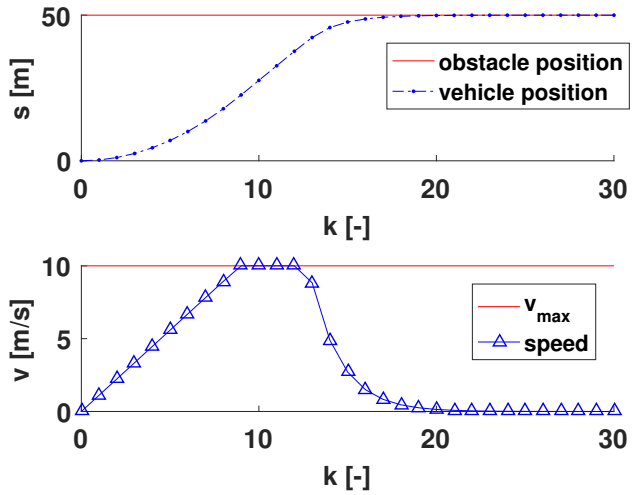


Figure 2.8: Simulation of one-step method with Ω and $s_{\text{obs}} = 50\text{m}$ obstacle.

Proposition 1. *If the objective function* $J = |v_{\max} - v\,(k+1|k)|$ *in MPC 2.2, then it results in an equivalent optimisation using* $J \equiv -s\,(k+1|k)$ *as an objective function.*

Proof. This is shown using the dynamics constraints that the optimisation is subject to

$$v\,(k+1|k) = v(k) + a(k|k)\;\delta t, \tag{2.31}$$

thus:

$$s\,(k+1|k) = \; s(k) + \frac{1}{2}(v(k) + v\,(k+1|k))\;\delta t, \tag{2.32}$$

resulting in:

$$J = |v_{\max} - v\,(k+1|k)| = \left| v_{\max} + \frac{2}{\delta t}\,(s(k) - s(k+1|k)) + v(k) \right|. \tag{2.33}$$

Furthermore, since the speed-operating region is $0 \leq v \leq v_{\max}$

$$J = |v_{\max} - v\,(k+1|k)| \equiv v_{\max} - v\,(k+1|k) \geq 0, \tag{2.34}$$

gives

$$J \equiv v_{\max} + \frac{2}{\delta t}\,(s(k) - s(k+1|k)) + v(k). \tag{2.35}$$

where the only optimised decision variable is $-s(k+1|k)$, a position term; the rest are constants resulting in mathematically equivalent optimisations subject to the original constraints. □

Figure 2.9 shows the v–s trajectory of Figure 2.8 with discrete states remaining within the bounds of Ω. In contrast, the continuous states can leave the set within an inter-sample

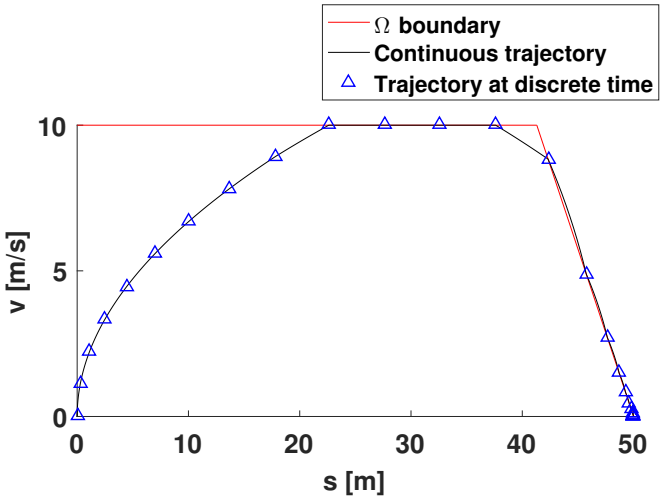


Figure 2.9: The v–s state trajectory in Ω-invariant set approaching a static obstacle.

period time but must return until the next discrete control time.

A trivial, sufficient condition requires that the initial states of the control must satisfy all constraints. In conclusion, the proposed control with Ω is recursively feasible for approaching static obstacles with constraints on the first predicted time step.

2.3.3.2 Numerical example: Parameter tests

Numerical tests were conducted with the aforementioned one-step control initiated from random x^0 states for different time headways to show feasibility. The region of initial states was chosen to result in varying points of contact on the rightmost slope of Ω. Counter-proof of recursive feasibility is shown by testing violating parameter choices against $\mathcal{T}$ control-invariant parameters in Figure 2.6. Simulations were run with different $t_\mathrm{h}/\hat{t}_\mathrm{h} = \{1, 0.99, 0.95, 0.7\}$ parameter ratios, where $\hat{t}_h := \underset{t_\mathrm{h}}{\mathrm{argmin}}(t_\mathrm{h},\ \delta t) \in \mathcal{T}$ (i.e. the lowest time headway resulting in the above-mentioned control-invariant Ω set for the operational limits at a given δt). This concludes, that $t_\mathrm{h}/\hat{t}_\mathrm{h} < 1$ cases are not control-invariant. As demonstrated in Figure 2.10, there was an increasing number of infeasible cases for smaller t_h parameters (at fixed δt), where the trajectories were shown until the point of infeasibility.

A visible trend is that only trajectories above a certain speed in contact with the slope may end up infeasible; this trend suggests a $\hat{v}_\mathrm{max}$ maximum speed limit, ensuring safety. Recall that, the inequalities (2.25) and (2.27) are conditions for Ω to be control-invariant (*Theorem 1*). The reformulation of the inequalities leads to the conditions for a safe maximum speed. Assume that all parameters are given and fixed, including t_h, with the exception of $\hat{v}_\mathrm{max}$, the safe speed condition is:

$$v \leq \hat{v}_\mathrm{max} = -a_\mathrm{min}\left(\frac{1}{2}\delta t + t_\mathrm{h}\right).$$

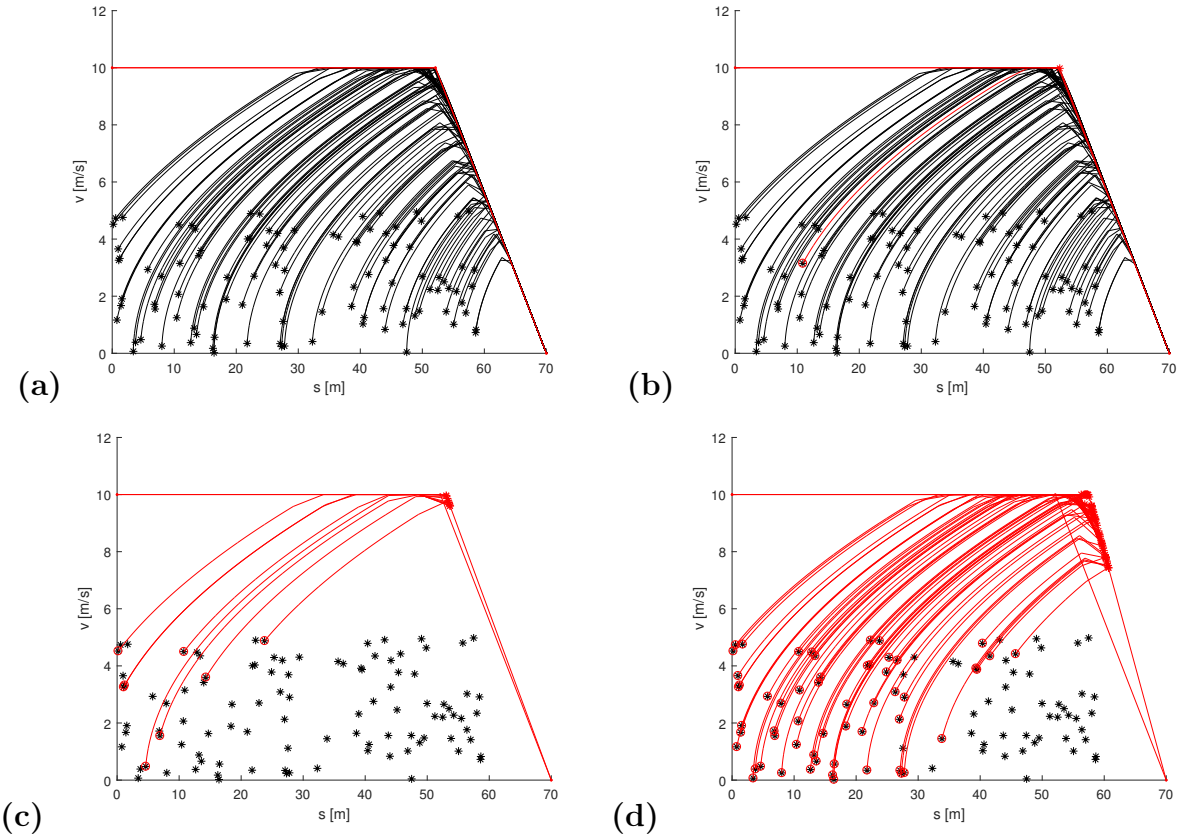


Figure 2.10: Simulations started from random initial states (black $*$) on **(a)**, **(b)**, **(c)** and **(d)** for $t_{\mathrm{h}}/\hat{t}_h = \{1,\ 0.99,\ 0.95,\ 0.70\}$ values, respectively. Simulation trajectories are shown with continuous black line, if feasible; if infeasible, simulation trajectories are shown in red until the last feasible point and marked with $(\circ)$ at the initial state. Only one simulation failed in **(b)**; in **(c)** and **(d)**, the number of infeasible simulations increases (only infeasible simulation are plotted in **(c)** and **(d)**).

2.3.4 Car-following

In car-following, two vehicles $\{p, q\} \in \mathcal{N}$ interact on a straight road where q is the leader vehicle and p is the follower vehicle. Alternatively, using ordered set representation $(p, q) \in \mathcal{N}$ would already encode the order of vehicles (or their indices) in the set. However, contrary to this section, orders can generally change; it could potentially be a temporary mode while following, for example, during merging or crossing. Commonly applied car-following policies in literature are summarised in [27].

$$s_p \leq s_q - h(x_p, x_q, u_p, u_q), \tag{2.36}$$

where $h(x_p, x_q, u_p, u_q)$ is the function governing the separation gap. Generally, the (gap) separation function in the inequality depends on the states and parameters of both the leader vehicle and the follower vehicle. In this work, the responsibility to remain at a safe distance from the leader vehicle belongs solely to the follower vehicle. The δt time delay of control and information propagation is accounted for using a spatio-temporal shift,

ultimately yielding:

$$s_p(k+1) \leq s_q(k) - t_{\mathrm{h}p} v_p(k+1), \tag{2.37}$$

where $t_{\mathrm{h}p}$ is the time headway-parameter of the follower vehicle. Thus, this case is a linear separation function h to prevent collisions with a time-delayed position of a moving obstacle treated as a stationary one.

This final inequality for car-following is arrived at alternatively by developing and extending the safe invariant set representation from Section 2.3.3. Later, this set representation formalism is used to create a more complex merging case with corner-cutting prevention, essentially guarding against inter-sample time violation of states in a manner similar to that of the spatio-temporal shift used in the above inequality.

Let invariant sets with vehicle-specific properties be as follows:

$$\Omega_n(s_{\mathrm{obs}}) := \Omega, \tag{2.38}$$

where subscript $n \in \mathcal{N}$ signifies, in general, the vehicle specific set parameters where applicable. This may include the headway time, maximum speed, and obstacle position relative to the vehicle, which, for convenience, is controlled through the argument.

Lemma 1. *: If $s_1 \leq s_2$, then $\Omega_n(s_1) \subseteq \Omega_n(s_2)$.*

Proof. This is true because the position argument sets the offset of the rightmost hyperplane constraint: $\Omega_n(s_1) = \{s, v \mid 0 \leq v \leq v_{\max n},\ s + t_{\mathrm{h}n} v \leq s_1 \leq s_2\}$. □

Theorem 2. *If $x_p \in \Omega_p(s_q)$, then this is control-invariant car-following under the parameter conditions from Theorem 1, where $\{p, q\} \in \mathcal{N}$ are vehicles moving on the same road section in the same direction and vehicle q precedes vehicle p.*

Proof. At $k = k_0$ initial time, $x_p(t_k) \in \Omega_p(s_q(t_k))$, where Ω_p is control-invariant for vehicle p according *Theorem 1.* Thus, the control sequence $\exists u_p \in \mathcal{U}$, such $x_p(t_k) \in \Omega_p(s_q(t_{k_0}))$, $\forall k \geq k_0$. Furthermore, from *Lemma 1*, $\Omega_p(s_q(t_{k_0})) \subseteq \Omega_p(s_q(t))$ for $\forall t \geq t_{k0}$ continuous time since $v_n \geq 0$, $\forall n \in \mathcal{N}$. Thus, $s_q(t_{k_0}) \leq s_q(t)$ according to the nominal dynamics assumption (2.8). □

In practice, the predictive controllers are not perfect; thus, $\{x_p(k),\ x_p(k+1)\} \in \Omega_p(s_q(k))$ is a stricter requirement incorporating the cautious control step leading to an increased gap between the follower vehicle and the leader vehicle with the increment being proportional to the speed and δt control period time.

Remark: As was mentioned previously, s position has the property of a reference arbitrary shifted by a constant. Thus, without a loss of generality, coordinate reference shifts and route offsets can be incorporated in the design of Ω_p (i.e. consider a projection

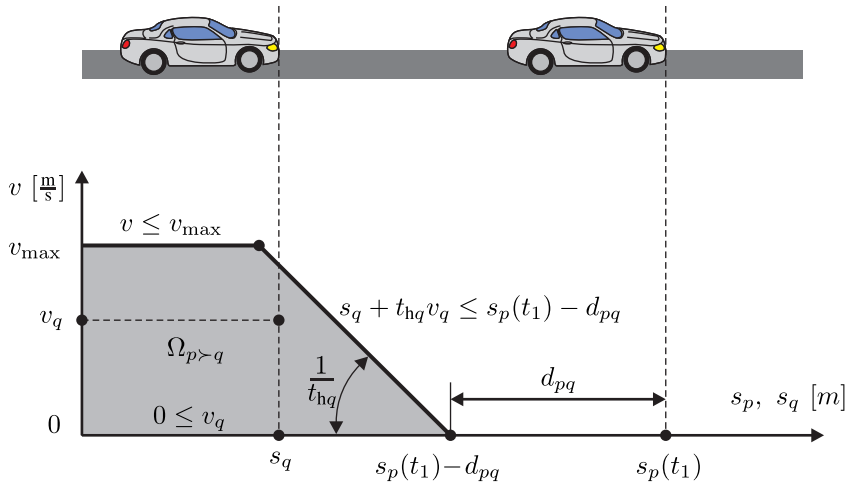


Figure 2.11: Car-following with $x_p \in \Omega_p(s_q)$.

function with the expression $x_p \in \Omega_p(\hat{s}_q)$ where $\hat{s}_q = s_q + L$ has been shifted by a constant L). This is a practical way to account for vehicle length as well as additional safety zones by reducing the available gap.

From the invariant set, the follower vehicle is granted the choice of a feasible stopping trajectory that makes no assumptions based on the leader vehicle's dynamics (see Figure 2.11). On the contrary, the Time-To-Collision (TTC) approaches allow close gaps of mere meters in platooning scenarios, even at high speeds, enabling vehicles to take advantage of reduced air drag and more closely packed traffic for greater road utilisation. However, these approaches are built on a solid knowledge of various factors, including dynamics and vehicle control, procedures for joining and leaving a platoon and trusted low-latency communication. In platoons, vehicle sequences are designed to provide safe, collision-free trajectories, even during emergency braking; they exploit the level of similarity and slight differences between vehicle capabilities through synchronised actions. It is easy to see how homogeneous platoon of vehicles simultaneously executing identical actions would result in the maintenance of any positive gaps, providing infinite TTC and safety. In realistic scenarios; however, any invalid assumption could result in a collision—even a simple deceleration would yield non-intuitive trajectory variations due to unmodelled and higher order terms (e.g. disc brake characteristics due to hydraulic actuation and delays, built-in pedal and calliper mechanisms, and tribological factors such as brake surface wear and fading). Ascertaining the appropriate separation between such trajectories is computationally difficult and would require comprehensive and continuous knowledge of such trajectories in real time, or even ahead of time.

2.3.5 Safe merging

A proposed framework for safe-merging builds on the static-obstacle approach and car-following modes. Consider the following aggregated set:

$$\Omega_{pq} := \{x_p, x_q | x_p \in \Omega_p(L_1) \vee x_q \in \Omega_q(L_2) \vee x_p \in \Omega_p(s_q + L_3) \vee x_q \in \Omega_q(s_p - L_4)\}, \quad (2.39)$$

where the two–two modes relate to the original obstacle $\mathcal{C}_{pq}$ and its four sides, as shown in Figure 2.4.

From *Theorem 1* and *Theorem 2*, it is safe to conclude that:

$$[x_p^{k_0}, x_q^{k_0}] \in \Omega_{pq} \Longrightarrow \exists u_p \in (u_p, u_q) \in (\mathcal{U}_p, \mathcal{U}_q),\ [x_p^k, x_q^k] \in \Omega_{pq},\ k \geq k_0,$$

which satisfies the $\left[x_p^k, x_q^k\right] \notin \mathcal{C}_{pq}$ condition.

However, this may be violated within the $t \in (t_k, t_{k+1})$ continuous time interval, resulting in collisions:

$$\forall [x_p^k, x_q^k], [x_p^{k+1}, x_q^{k+1}] \in \Omega_{pq},\ \nRightarrow [x_p(t), x_q(t)] \notin \mathcal{C}_{pq},\ t \in (t_k, t_{k+1}).$$

An obvious case of this phenomenon is when a leader vehicle and a follower vehicle switch roles between two discrete time steps; this is referred to as corner-cutting, a characteristic issue of discrete-time constrained problems and non-convex obstacle avoidance.

2.3.5.1 Corner-cutting prevention

The disadvantageous cut of the solution trajectory into the obstacle region (i.e. vehicle collision) in continuous time may happen in cases were trajectories are constrained only at discrete time steps, mainly around constraints defining obstacle corners [63]. This phenomenon can be avoided through corner-cutting prevention—also known as inter-sample avoidance—by requiring a handover between discrete time steps and modes. The enforcement of a temporal handover between discrete time steps is often implemented in MIP formulation either through logical constraints between binaries [50] or by duplicating and temporally shifting the modes (active set of constraints) [70].

Let the safety-preserving modes from (2.39) be named as:

$$\begin{aligned} &\text{m1}: && x_p \in \Omega_p(L_1), \\ &\text{m2}: && x_q \in \Omega_q(L_2), \\ &\text{m3}: && x_p \in \Omega_p(s_q + L_3), \\ &\text{m4}: && x_q \in \Omega_q(s_p + L_4). \end{aligned} \quad (2.40)$$

Each mode has an alternative interpretation of a positive control-invariant safe set of states. A state is able to remain in any mode set safely and indefinitely if it is already

part of that set and chooses to remain as such (see positive control invariance).

Now let the following sequence show with (T: true; −: false) when different modes are active and set constraints are satisfied:

	$k-3$	$k-2$	$k-1$	k	$k+1$	$k+2$	$k+3$
m1	T	T	T	T	−	−	−
m2	T	T	−	−	−	−	−
m3	−	−	−	T	T	T	T
m4	−	−	−	−	−	−	−

The states must be safe at any k discrete time; thus, at least one mode must be active at each time for all pairwise vehicle-interaction obstacles. This consideration relates to the separating hyperplane theorem [60].

Furthermore, both modes have to be active (true) at the moment of handover for corner-cutting prevention between switching modes, as shown in the above example at k.

Suppose the new set $\Omega_{pq}^{\text{ccp}} := \Omega_{pq}(+\text{binary logic})$ is designed to be control-invariant and collision-free with respect to continuous time. This can be achieved by implementing the logic of two methods from [50, 70], which are compared in [63] using the big-M approach. The advantage of using the approach in [70] with duplicated and temporally shifted original constraints stems from the improved scaling and less sensitive numerics in binary logic. As such, it is the preferred implementation later in this work.

For example, the transition m1 $\rightarrow$ m3 at k represents the entrance into car-following mode; aforementioned logic of corner-cutting prevention would yield a smaller set of states:

$$x_p(k) \in \Omega_p(L_1) \cap \Omega_p(s_q + L_3),$$

which can cause additional artefacts when the state evolution is large between the time step, especially for long periods times. Mitigation for these artefacts using shadow-region description is considered to be a less strict formulation [56, p. 65], in which hyperplanes separating the next state are generated as a function of the linear visibility from the previous state. However, this would remove the convex structure of the problem when the binaries are fixed.

2.4 Feasible paths to goal

This section discusses the possibility and practicality of feasible and non-feasible paths to reach vehicle goals in the face of obstacle shapes, positions and, in cases of multiple obstacles, their constellations. The obstacles from the junction interaction collision sets are convex; however, the exclusion of such sets generally makes their optimisation non-convex, imposing the risk of local minima or unattainable goal positions. Furthermore,

on account of the finite-length planning horizons, an earlier decision of vehicle orders may later impede the progress of vehicles toward their goals and trap the solution trajectory. For examples, in Figure 2.12 **(a)**, **(c)** and **(e)**, see how the B paths around the obstacles end up trapped and fail to reach the goal position for one or both vehicles (without reversing). These situations represent cases in which both the vehicle goal position and the vehicle interaction (collision set) act as an obstacle. Thus, a constellation of multiple obstacles, even if the individual elements are convex, is inherently able to form traps.

In Figure 2.12 **(b)**, **(d)** and **(f)**, in contrast, either vehicle order gives way to a solution to reach the vehicle goals—no obstacles overlap. In this work, the examples address cases in which the vehicle order has a neutral effect on goal reachability. In most cases, goal reachability issues are not present in simple junction simulations, as vehicles never intend to park in the vicinity of the junction; rather, they aim to leave the area of junction (i.e. the goals are far, usually outside of the simulated area).

However, when multiple vehicles interact, the multiple collision sets may form non-convex obstacles with traps in higher-dimensional state space from the amalgam of convex pair-wise collision sets (represented in the higher dimensional state space). Following the reasoning in [57], the global solution is attainable when the system has a sequence of vehicles represented on a priority graph without loops [30]. The intuitive way to show that junction simulations without cyclic priority graphs are able to reach their solution is phrased in [57]. Namely, if the system (junction simulation) always has at least one vehicle that can freely move forward and either reach its non-obstructing goal or leave the system, the system eventually clears up and all vehicles can eventually reach their goal or leave the system.

Remark: Here the above idea in [57] is rephrased. If a feasible path is transferring the vehicles to their goal position with non-changing system parameters and obstacles, then $\forall t_k, \exists n \in \mathcal{N}$, such that vehicle n had a free choice to move, $v_n(t_k) > 0$, when $s_n(t_k) < s_{n\ G}$.

In practice, positive vehicle speeds can be indicators of progress and liveliness though asymptotic speed profiles for deceleration in a trap would satisfy this condition.

The convexity of the obstacles, even in a single pair-wise collision set case, is not satisfying as obstacle avoidance from certain state configurations can only be done through reversal, which is against the non-negative speed constraint. For example, imagine a merging or box obstacle (Figure 2.12) with, in place of the bottom-left right angle, an obtuse one; the shape would still be convex but the feature would act as a trap without the ability to reverse. Thus, all collision sets are required to be convex (satisfied by the supporting hyperplane description) and have all obstacle boundaries in any point allow at least one vehicle to have positive speed towards its goal:

$$\max\left(\frac{\partial g_i(s_p, s_q, v_p)}{\partial s_p}, \frac{\partial g_i(s_p, s_q, v_p)}{\partial s_q}\right) > 0, \quad \forall i = \{1 \dots N_{pq,h}\}, \tag{2.41}$$

where the i-th hyperplane is $g_i \leq 0$, and the number of hyperplanes that define the collision set (obstacle) is $N_{pq,h}$.

Remark: this condition is true for single or separate obstacle cases; as mentioned, multiple convex obstacles are still able to form a non-convex union obstacle when overlaid. For example, this can happen when vehicles are trapped in a gridlock prone intersection, reaching a local minimum of the problem; the obstacle in such a case forms a pocket in the higher-dimensional configuration space and traps the system state once it has entered into the pocket. Intuitively, in the higher-dimensional space, it is necessary for at least one of the vehicles to be able to move at any obstacle-free point in the configuration space; it is generally possible to formulate a condition similar to (2.41). In this work, only atomic junction blocks are examined, as they are simple enough to be not prone for such structural gridlocks.

Let the combined obstacle set be defined by the union set of collision positions, where each 2D collision set is extruded to the high-dimensional space of the total number of agents. First, by redefining $\mathcal{C}_{pq}$, it would cast itself to the appropriate dimensional space of $\mathbb{R}^N$, $N > 2$:

$$\mathcal{C}_{pq} = \left\{ \boldsymbol{s} \in \mathbb{R}^N \middle| \, [s_p, v_p, s_q, v_q] \in \mathcal{C}_{pq}, v_p = v_q = 0 \right\} \tag{2.42}$$

This extrudes the pairwise collision sets into the higher dimension. Following this, let the overall obstacle be the union of all pairwise collision sets:

$$\mathcal{C} = \bigcup_{\forall p \in \mathcal{N}, q \in \mathcal{N}, p \neq q} \mathcal{C}_{pq} \tag{2.43}$$

then requiring

$$\{s^k, s_G\} \notin \mathcal{C} \tag{2.44}$$

with the previous requirement on hyperplanes, (2.41), it is ensured that a trajectory exists between current s^k and goal position s_G. The completion time of this trajectory may, however, depend on vehicle-specific dynamics. Consider the worst-case scenario: the trajectory exists only when the previously moving vehicle p has zero speed. While in motion, the safety constraints allow for a full stop; however, if asymptotic convergence is assumed, the convergence time may be ∞. This would mean that, in some situations, the vehicles are so tightly packed that they temporarily block one another, which, in a practical sense, is still solvable. The model considers a certain safety length over the physical vehicle length. When slow convergence occurs, it may be beneficial to give up a portion of the safety distance at low speeds to allow for reasonable vehicle progress. Alternatively, a better suited controller could be switched to and optimise trajectory with high-fidelity models for close vehicle manoeuvres (complete motion planning). It may

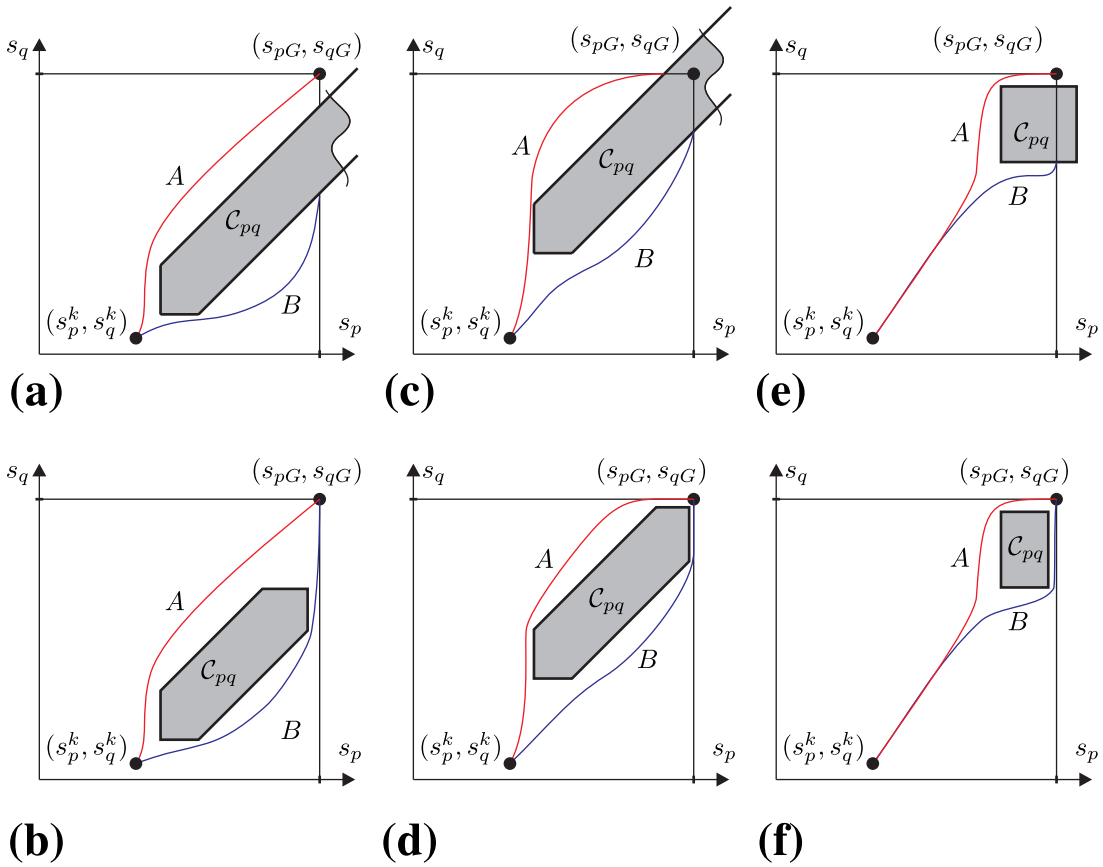


Figure 2.12: Possible obstacle evasions ($\mathcal{C}_{pq}$ collision sets) for a vehicle pair with goal position s_{pG}, s_{qG} and two possible trajectories A, B corresponding to the two vehicle orders. Car-following after merging is shown in **(a)** and **(c)**; merging, car-following and diverging is shown in **(b)** and **(d)** while a cross-junction-crossing scenario is shown in **(e)**, **(f)**. The cases of **(a)**, **(c)** and **(e)** show an unreachable goal position for ordering choice B, where one vehicle obstructs the other. Both vehicle orders in **(b)**, **(d)** and **(f)** are non-obstructing in terms of goal reachability. Further considerations for the feasibility of similar problems can be found in [31].

also be worthwhile to virtually define the goal position farther ahead to reach finite time convergence with some negligible speed left at the moment of arrival.

2.5 Mixed-integer model predictive control

The MPC optimisation can be defined over decision variables of continuous and discrete domains. Generally, it is known as hybrid-MPC when both types of decision variables are used; when the discrete states are integers, it is known as mixed-integer MPC. The MPC theory is introduced well in [8] and demonstrated in control applications in [61]. Discrete-decision variables can be used to implement piecewise affine (PWA) functions and discrete events. Moreover, discrete states can be used to implement discrete rules and relaxations; in this work, the discrete-decision variables are binaries $b \in \{0,\ 1\}$ used for constraint relaxation (i.e. turning on and off specific traffic rules following safe logic).

The MPC problem with a horizon length of N_p and multiple agents results in the following optimisation problem:

MPC-2.3

$$
\begin{aligned}
J^*_{\text{MPC-2.3}} = \min \; & \sum_{n\in\mathcal{N}} \sum_{j=0}^{N_\mathrm{p}-1} l_n\left(x_n\left(k+j+1|k\right), u_n\left(k+j|k\right)\right) \\
\text{s.t.} \quad & \forall j \in \{0 \dots (N_\mathrm{p}-1)\},\ \forall n \in \mathcal{N}: \\
& x_n\left(k|k\right) = x_n\left(k\right) \\
& x_n\left(k+j+1|k\right) = A\ x_n\left(k+1|k\right) + B\ u_n\left(k+1|k\right) \\
& x_n \in \mathcal{X}_n \\
& u_n \in \mathcal{U}_n \\
\text{and } & \forall p, q \in \mathcal{N},\ p \neq q,\ C_{pq} \neq \emptyset: \\
& [x_p, x_q] \in \Omega^{\text{ccp}}_{pq},
\end{aligned}
$$

where the operational constraints for the states are encoded in $\mathcal{X}_n$ (e.g. speed range) and for control inputs $\mathcal{U}_n$ (e.g. acceleration range). The cost function is chosen to be a simple quadratic 'running-cost', which is discussed in depth later in the work. The stage costs in the cost formulation are as follows:

$$
l_n\left(x_n, u_n\right) = Q\left(v_n - v_{\mathrm{d}\ n}\right)^2 + R\ u_n^2, \tag{2.46}
$$

where the quadratic cost weights are Q and R and the agent-wise desired speed is $v_{\mathrm{d}\ n}$. The logic is contained in Ω^{ccp}_{pq}, which selects the necessary constraints to realise the control-invariant sets. The logic is based on binary decision variables, a sub-class of integers rendering the overall problem as a mixed-integer program (MIP). The original, non-negative speed assumption reappears in Ω^{ccp}_{pq} definitions and is already included in the domain $\mathcal{X}_n$ of states. Without duplicating the two speed limits, only the hyperplane separating x states and the separating hyperplane theorem constraint needs to be added. Let this constraint for the modes in general for vehicle $n \in \{p, q\}$ be represented as:

$$
g(x_n, s_\text{obs}; t_\text{hn}) \leq 0.
$$

The big-M relaxation method was chosen to relax the linear inequality constraints because it is a straightforward formulation, does not add additional non-convexities to the problem, and does not increase the problem size [38]. Through this relaxation, a constraint is practically activated or deactivated as:

$$
g(x_n, s_\text{obs}; t_\text{hn}) \leq M\ b,
$$

where a sufficiently large constant is M and a binary decision variable is $b \in \{0,\ 1\}$. In fact, according to [8], M is to be chosen as:

$$M \triangleq \max_{x \in \mathcal{X}} g(x, s_{\text{obs}}; t_{\text{hn}}). \tag{2.47}$$

In practice, only some of the examples shown have fully bounded $\mathcal{X}$ domains, for which M can be chosen to respect the boundaries. If this is not possible, M is chosen to be non-restrictive in the simulations, given the practical sizes of the numerical test region as well as the distance covered over the horizon lengths. The tightness of the constraints in the big-M formulation strongly depends on the size of M; consequently, a smaller M is better, though the formulation is still weak by nature when compared to other relaxation methods [38].

Moreover, the formulation used is not well-posed, according to [8], because the b binary states are not unique indicators of the violation state of the relaxed inequality (i.e. when $b \equiv 1$, the respected inequality may be either satisfied or in a violated state). This does not compromise the relaxation formulation; however, it may impose a performance trade-off between well-posed binaries with extra overhead and better branch-and-bound search speeds.

In practice, the four modes can be simplified as four linear inequalities (hyperplanes) and the connecting logic:

$$g_1(x_p, L_1; t_{\text{hp}}) \leq M\ b_1 \tag{2.48a}$$
$$g_2(x_q, L_2; t_{\text{hq}}) \leq M\ b_2 \tag{2.48b}$$
$$g_3(x_p, s_q + L_3; t_{\text{hp}}) \leq M\ b_3 \tag{2.48c}$$
$$g_4(x_q, s_p + L_4; t_{\text{hq}}) \leq M\ b_4 \tag{2.48d}$$
$$\sum_{m=1}^{4} b_m \leq 3. \tag{2.48e}$$

Note that when $b_m \equiv 1$, the corresponding inequality is relaxed. Moreover, (2.48e) inequality ensures that at least one of the four modes is active (not relaxed), which represents the logic of the separating hyperplane theorem. Thus, the above set of inequalities in (2.48a)-(2.48e), and (2.9) encode Ω_{pq}. Recall that corner-cutting between two time-steps k and $k{+}1$ can still happen when the subsequent states $[x_p^k, x_q^k] \in \Omega_{pq}$ and $[x_p^{k+1}, x_q^{k+1}] \in \Omega_{pq}$. Thus, following the method [63, 70] as discussed in Section 2.3.5.1 the corner-cutting pre-

vention employs binaries and yields:

$$g_1(x_p(k), L_1; t_{\mathrm{h}p}) \leq M\ b_1(k) \tag{2.49a}$$
$$g_2(x_q(k), L_2; t_{\mathrm{h}q}) \leq M\ b_2(k) \tag{2.49b}$$
$$g_3(x_p(k), s_q(k) + L_3; t_{\mathrm{h}p}) \leq M\ b_3(k) \tag{2.49c}$$
$$g_4(x_q(k), s_p(k) + L_4; t_{\mathrm{h}q}) \leq M\ b_4(k) \tag{2.49d}$$
$$\sum_{m=1}^{4} b_m(k) \leq 3 \tag{2.49e}$$
$$g_1(x_p(k+1), L_1; t_{\mathrm{h}p}) \leq M\ b_1(k) \tag{2.49f}$$
$$g_2(x_q(k+1), L_2; t_{\mathrm{h}q}) \leq M\ b_2(k) \tag{2.49g}$$
$$g_3(x_p(k+1), s_q(k+1) + L_3; t_{\mathrm{h}p}) \leq M\ b_3(k) \tag{2.49h}$$
$$g_4(x_q(k+1), s_p(k+1) + L_4; t_{\mathrm{h}q}) \leq M\ b_4(k), \tag{2.49i}$$

where the same binaries are enforced for the next time step for the mode constraints. This satisfies the logic between the time steps outlined in the requirements of $\Omega_{pq}^{\mathrm{ccp}}$, providing the safe transition between modes and naturally increasing the number of inequalities from five to nine.

2.5.1 Robustness for sudden stop events

The further extension of spatio-temporal constraints provides a stricter formulation and adds robustness against sudden stop events that may occur during low-speed urban driving or as a result of general mismatch between predicted and actual states. Suppose that obstacle position $s_q(k+1)$ in (2.49h) is replaced with $s_q(k)$. When this change is applied to both (2.49h) and (2.49i), it results in:

$$g_3(x_p(k+1), s_q(k) + L_3, t_{\mathrm{h}p}) \leq M\ b_3(k) \tag{2.50}$$
$$g_4(x_q(k+1), s_p(k) + L_4, t_{\mathrm{h}q}) \leq M\ b_4(k), \tag{2.51}$$

where the new position of the follower vehicle must respect the obstacle position that is determined by the current position of the leader without making any assumptions about the vehicle dynamics of the leader.

In this case, observe that from the four modes for m1 and m3, $x_p^k \in \Omega_p(s_{\mathrm{obs}}(k))$ and $x_p^{k+1} \in \Omega_p(s_{\mathrm{obs}}(k))$. This means that vehicle p is not only capable of controlling its future states to remain in the control-invariant set but is also required to remain in that set for the next time step. For example, this formula decouples the time-dependent terms of the leader vehicle dynamics, treating it like a true static obstacle between time steps, which results in a more conservative but safe set against sudden stops referred to by the

combined $\Omega_{pq}^{\text{ccp,st}}$:

$$\Omega_{pq}^{\text{ccp,st}}(k) := \{x_p^k, x_p^{k+1}, x_q^k, x_q^{k+1}, b_i^k,\ i = 1, \dots 4 | (2.49a)\text{–}(2.49g), (2.50), (2.51)\} \quad (2.52)$$

Remark: Suppose the system is represented by only (2.49e)–(2.49g), (2.50) and (2.51). This would, for example, simplify in m1 and m3 modes to $x_p^{k+1} \in \Omega_p(s_{\text{obs}}(k))$ constraint set (with only the states of the next time step x_p^{k+1}, allowing for a larger attractor set for x_p^k initial states), increasing the noise tolerance and recovery capabilities of the control without explicitly needing to calculate such an attractor set. However, the corner-cutting prevention logic expects constraint feasibility checks for both discrete times and requires $x_p^k,\ x_p^{k+1}$ states, which is easier to be fulfilled by the full $\Omega_{pq}^{\text{ccp,st}}(k)$ set.

2.6 Numerical tests: Merging with two vehicles

Feasibility tests for Y-junction merging were conducted with two identical vehicles for different t_{h}–δt parameter pairs. Initial speeds were set to $v^0 = v_{\max} = v_{\text{des}} = 10$ ms^{-1}. Initial positions were chosen randomly to provide conflicting arrivals at the junction; the chosen positions were far enough from the junction to respect initial feasibility and non-interfering predictions (i.e. $s_p^0 \leq L_{pq}^1 - (t_{\text{h}p} + N_{\text{p}}\delta t)v^0$). The following parameters were chosen: time horizon $N_{\text{p}} = 5$, $a_{\min} = -9.81/2$ ms^{-2} and $a_{\max} = 3$ ms^{-2}; for simplicity, $s_{\text{off1}} = s_{\text{off2}} = 0$ m; $L_1 = L_2 = 4$ m.

MPC-2.4

$$J_{\text{MPC2.4}}^* = \min \sum_{n\in\mathcal{N}} \sum_{j=0}^{N_{\text{p}}-1} l_n\left(x_n\left(k+j+1|k\right), u_n\left(k+j|k\right)\right) \quad (2.53a)$$

$$\text{s.t.} \quad \forall j \in \{0 \dots (N_{\text{p}}-1)\},\ \forall n \in \mathcal{N}:$$

$$x_n\left(k|k\right) = x_n\left(k\right)$$

$$x_n\left(k+j+1|k\right) = A\ x_n\left(k+1|k\right) + B\ u_n\left(k+1|k\right)$$

$$x_n \in \mathcal{X}_n$$

$$u_n \in \mathcal{U}_n$$

$$\text{and } \forall p, q \in \mathcal{N},\ p \neq q,\ C_{pq} \neq \emptyset:$$

$$[x_p, x_q] \in \Omega_{pq}^{\text{ccp,st}}.$$

The cost function to be minimised is set to quadratic speed error penalisation without acceleration penalty terms ($R = 0$), as this allows for better state mapping due to the

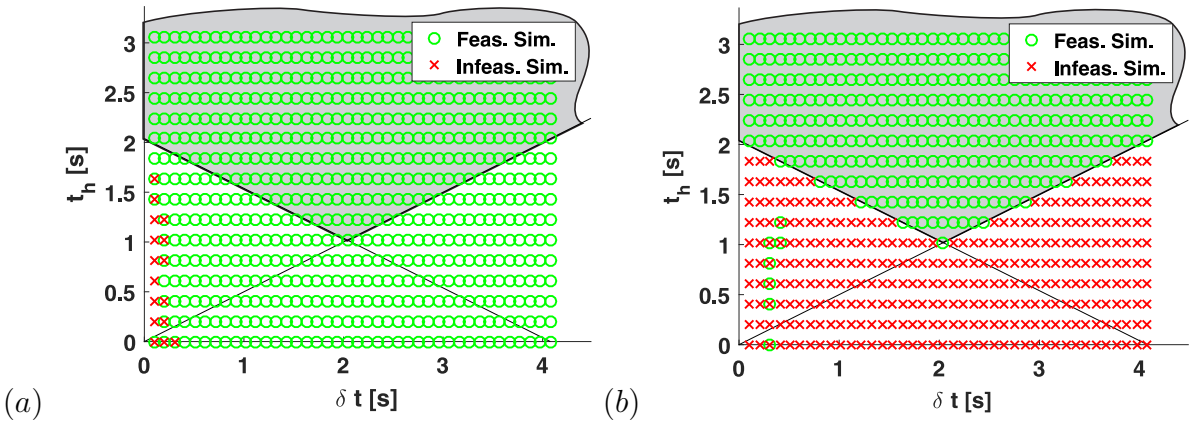


Figure 2.13: MPC merging simulations for parameters t_{h}–δt; MPC merging simulations for parameters t_{h}–δt with a sudden stop of the leader vehicle. [6]

greedier speed policy and, thus, a better map of potentially infeasible cases.

$$J(v_1, v_2) = \gamma\, Q\,(v_1 - v_{\mathrm{des}})^2 + (1-\gamma) Q\,(v_2 - v_{\mathrm{des}})^2\,, \tag{2.54}$$

where a single relative weighting factor $\gamma \in (0,1)$ is introduced for simplicity (with $w_1 = \gamma$, $w_2 = 1-\gamma$). Figure 2.13 **(a)** displays 150 simulation runs for each pair of parameters and indicates where problems remained feasible until the end of the simulation or became infeasible due to constraint violations. The results show that for low-δt choices, the used number of time-horizon steps led to infeasible cases; for sufficiently high-δt choice (long horizons), all simulations were ultimately feasible, as predicted by [2].

Figure 2.13 **(b)** displays results from another simulation experiment in which the leader vehicle suddenly stops at the merging junction. This disturbance was achieved by setting the speed of the leader vehicle to 0 ms^{-1} at its previous position (with no position state evolution) after passing the junction. The only parameters, that were chosen in line with the conditions defined in *Theorem 1*, were able to obtain positive control invariance for the vehicles and result in recursive feasibility. However, MPC problems without positive control-invariant sets fail to stop for harsh disturbances when the previous prediction is not perfectly followed.

In an experiment for three different γ values, their critical initial positions were identified by s_1^0 and s_2^0. In order to simplify the test, s_1^0 initial position was kept constant throughout all simulations while the spacing between the vehicles was controlled by the relative initial position gap $\Delta d := s_2^0 - s_1^0$. Figure 2.14 presents the state trajectories taken; cases in which vehicle 1 passed the junction first are shown in blue while those in which vehicle 2 passed the junction first are in red. For a given γ prioritisation weight, a critical Δd value can be found where it is more beneficial to change the vehicle order.

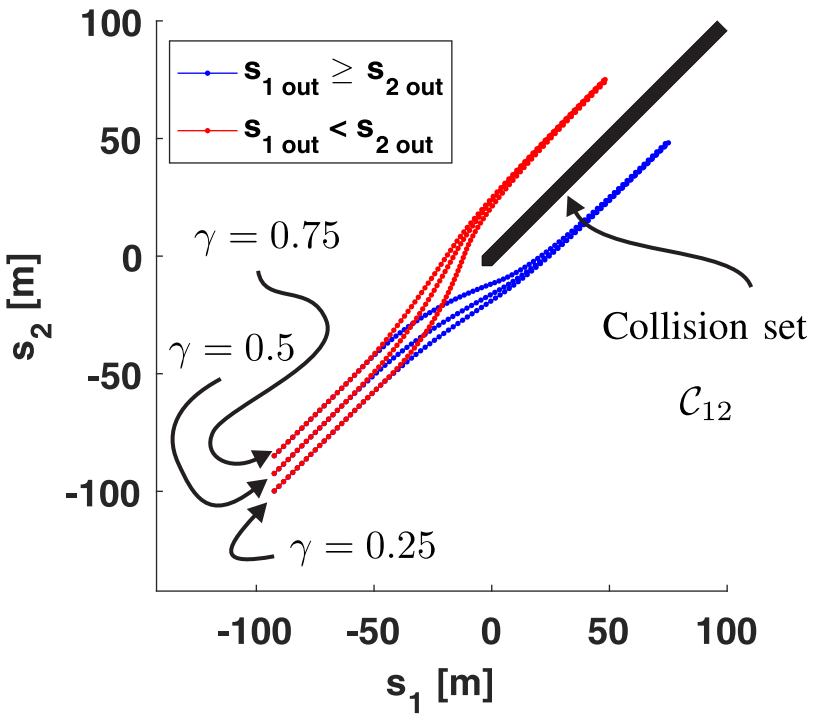


Figure 2.14: Merging for three γ and critical Δd switching values with δt = 0.2 s, t_{h}=2.1 s, $N_{\mathrm{p}} = 25$, Q=1, R≃5.1. [6]

2.6.1 Decision graph

Simulations were conducted for a range of weightings $\gamma \in (0, 1)$ and starting-position offsets Δd. Figure 2.15 **(a)** presents four quadrants. The second and fourth quadrants depict the cases in which first-come-first-served outcomes were observed; in these regions, prioritisation enforces the initial order, meaning that if a vehicle is ahead of another vehicle, it passes the junction first. However, with γ weighting in the first and third quadrant, the vehicle orders could be changed, resulting in a consistent prioritisation effect. A distinct switching line can be obtained by connecting the adjacent critical Δd values on the decision graph. This line separates the cases in which one vehicle finished first from those in which the other did. The switching line at neutral prioritisation, $\gamma = 0.5$, was at $\Delta d = 0$ m. Figure 2.15 **(b)** shows the effect of high-δt period times for the same $N_{\mathrm{p}}\delta$t=5 s horizon time; it results in a distorted, asymmetric decision graph due to the effect of discretisation and corner-cutting prevention. The longer the time steps, the more some cases must be constrained to satisfy the overlaying constraints between the mode switches, resulting in a cruder line between the ranges.

2.6.2 Numerical test: Symmetric decision graph

The decision split curves of the tests described in Section 2.6.1 are presented on decision graphs in Figure 2.15; they have asymmetrical properties (i.e. mirroring the curve by the axes at $\Delta d = 0$ and $\gamma = 0.5$ concludes that the mirrored curve does not line up with

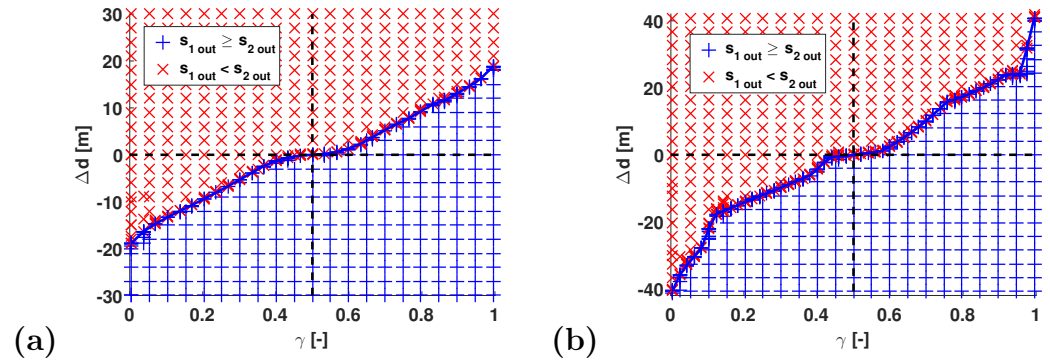


Figure 2.15: **(a)** Priority graph with FCFS regions in 2nd and 4th quadrants and shifted priority due to the γ objective weighting in the 1st and 3rd quadrants, δt = 0.2 s, t_h = 2.1 s, $N_\mathrm{p} = 25$, Q=1, R≃5.1; **(b)** Priority graph with artefact from discretisation, δt = 2.5 s, t_h = 2.1 s, $N_\mathrm{p} = 2$, Q=1, R ≃5.1. [6]

its original curve). The reason behind this asymmetry was found to be the choices of initial positions, which were only sampled from a line parallel to the axis of s_2. The initial position choices were fixed for s_1^0 and changing for s_2^0, which is shown for three pairs of trajectories in Figure 2.14. As the problem description, obstacles and vehicle properties are symmetric, the decision graph is expected to be symmetrical as well. Figure 2.16 shows the Δd choice of axis at d_e distance from the edge of the $\mathcal{C}_{12}$ collision set. This new adjusted axis has the same symmetric properties as the obstacle while maintaining the Δd initial gap position parametrisation.

Simulation test parameters were as follows: $\delta t = 0.5$ s, $t_\mathrm{h} = 0.78$ s, $N_\mathrm{p} = 0.5$, $v_\mathrm{max} = v_\mathrm{des} = 5$ ms^{-1}, $a_\mathrm{max} = 9.81/4$ ms^{-2}, $a_\mathrm{min} = -9.81/2$ ms^{-2}. In Figure 2.17, a decision split curve was identified at each slice for different d_e distances. Under these conditions, each decision curve has symmetric properties over γ, relative to the prioritisation axis. At low d_e distance, the feasible decision curves are close to being horizontal and γ has little effect. The effect grows as distance from the obstacle increases and more space is available for manoeuvring. After reaching a given limit d_e distance, the distance corresponding to the finite horizon has no direct contact with the obstacle and, thus, will not significantly change further. However, a periodic ripple can be seen on the curves, indicating that the decision process is sensitive to the level of discretisation and the inter time step phase when the manoeuvre is initiated; this explains the asymmetric results of the method shown in Section 2.6.1.

Figure 2.18 shows the aggregated distances travelled by both vehicles in these examples.

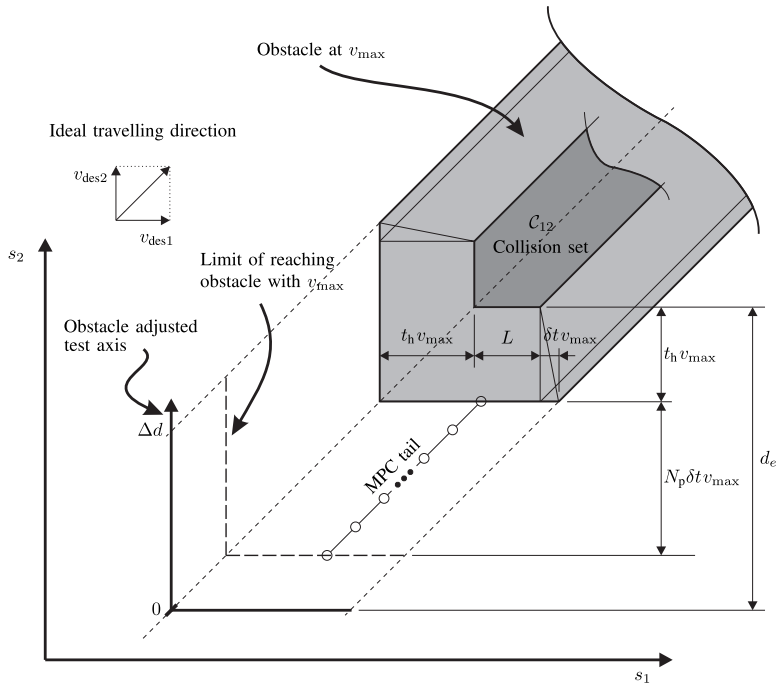
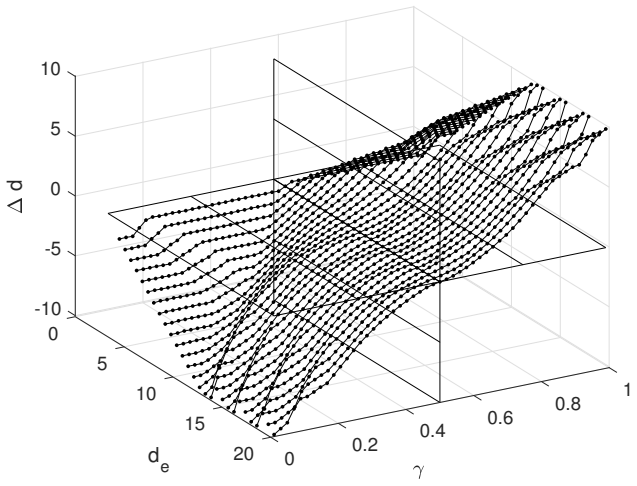
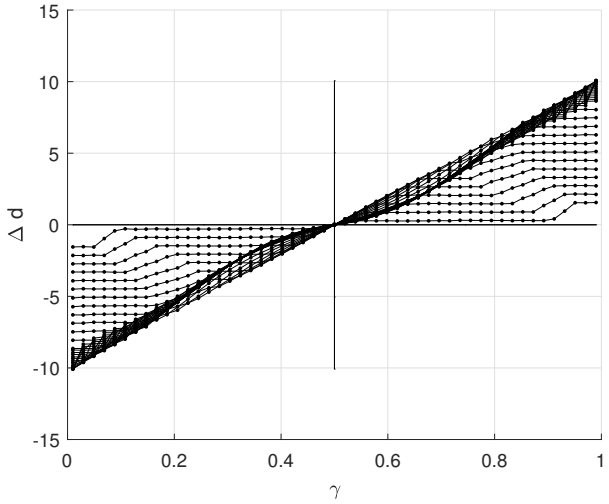


Figure 2.16: Schematic of tests with an adjusted axis (respecting obstacle symmetricity) for choice of initial vehicle position.

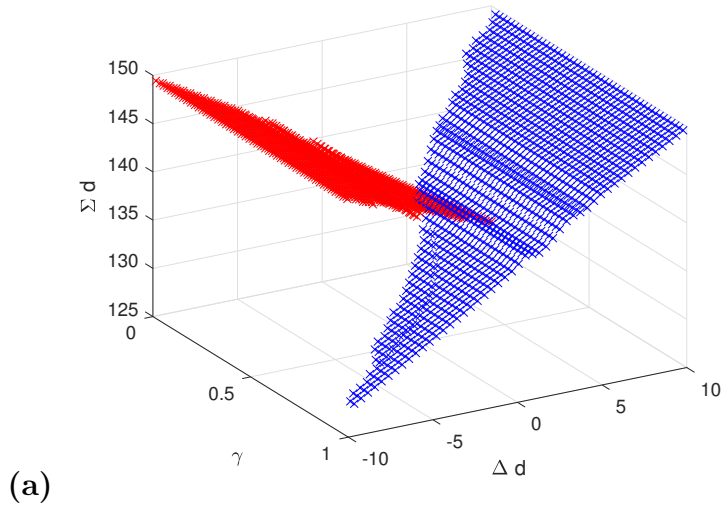


(a)

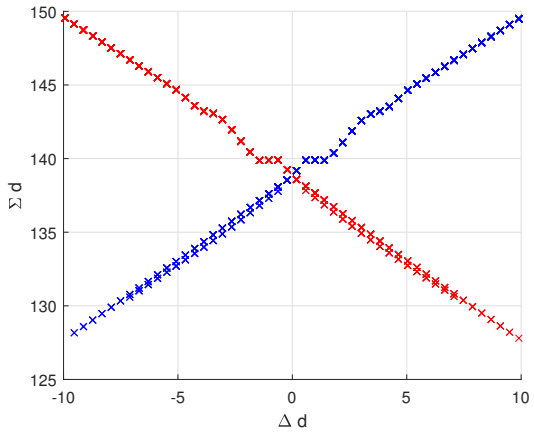


(b)

Figure 2.17: In **(a)**, the feasible-decision critical Δd curves are shown for different d_e distances from the obstacle and γ prioritisations. **(b)** shows the side view of **(a)**. The further the initial vehicle positions (starting with $v_{\max}$) are from the obstacle, the more significant the influence of relative prioritisation γ is over Δd while the curves remain symmetric; after a given limit distance, the curves display periodicity.



(a)



(b)

Figure 2.18: The full distance travelled Σd by both vehicles summed across the all simulations, at $d_e = 20$ m on **(a)** a 3D representation with the γ axis; **(b)**, a side view of **(a)**, shows little dependency on the γ parameter.

2.7 Numerical tests: Four lanes and vehicles merging

This section presents numerical tests of a scenario in which four vehicles merge from four different lanes (see Figure 2.19).

Vehicular control for four-lane merging is detailed in MPC 2.5, where the merging interactions are collected in $\mathcal{N}_{\mathrm{MI}}$ with vehicle pair tuples $\{p, q\} \in \mathcal{N}_{\mathrm{MI}}$. The number of tuples in $\mathcal{N}_{\mathrm{MI}}$ is N_{MI}, which is 'four choose two' for this case, $N_{\mathrm{MI}} = 6$ because two element subsets (vehicle pairs) are chosen from the four-element vehicle set $\mathcal{N}$. The parameters in the example were chosen as follows: δt=1 s, t_{h}=2.1 s, $N_{\mathrm{p}} = 5$, the vehicle weights are $w_{n=1\ldots4}$=[0.25, 0.05, 0.1, 0.6] for $n = 1 \ldots 4$ vehicle indices, Q=1, $R \simeq 5.1$ and $s_{\mathrm{off}n=1\ldots4}$=0 with vehicle length $L = 4$ m, making the obstacle geometries identical with $L_1 = L_2 = L_3 = L_4 = -L$. The desired speed is $v_d = 9$ ms^{-1} while the maximum speed is $v_{\max} = 10$ ms^{-1}. The position evolution of the four vehicles is shown in Figure 2.20; their speeds are shown in Figure 2.21. It is verified by the simulated trajectories, that by setting relative priorities for the vehicles (ω_n), they act in a cooperative manner. In the example, high priority vehicles merge first at the junction before accelerating or decelerating back to their desired speed. In Figure 2.21, the first vehicle to merge is shown to increase its speed over its desired speed in order to ease the control actions of the vehicles behind it.

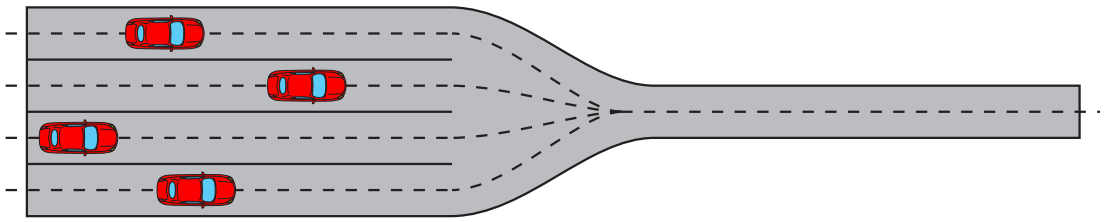


Figure 2.19: Four separate lanes merging to one lane

MPC-2.5

$$J^*_{\text{MPC-2.5}} = \min \sum_{n\in\mathcal{N}} \sum_{j=0}^{N_\text{p}-1} \omega_n \; l_n \left(x_n \left(k+j+1|k \right), u_n \left(k+j|k \right) \right) \tag{2.55a}$$

s.t. $\forall j \in \{0 \dots (N_\text{p}-1)\}, \; \forall n \in \mathcal{N}$:

$$x_n \left(k|k \right) = x_n \left(k \right)$$

$$x_n \left(k+j+1|k \right) = A \; x_n \left(k+j|k \right) + B \; u_n \left(k+j|k \right)$$

$$x_n \in \mathcal{X}_n$$

$$u_n \in \mathcal{U}_n$$

and $\forall \{p, q\} \in \mathcal{N}_\text{MI}$:

$$s_p(k+j|k) + t_\text{h} v_p(k+j|k) \leq L_1 + M \; b_{pq\;1}(k+j|k)$$

$$s_q(k+j|k) + t_\text{h} v_q(k+j|k) \leq L_2 + M \; b_{pq\;2}(k+j|k)$$

$$s_p(k+j|k) + t_\text{h} v_p(k+j|k) \leq s_q(k+j|k) + L_3 + M \; b_{pq\;3}(k+j|k)$$

$$s_q(k+j|k) + t_\text{h} v_q(k+j|k) \leq s_p(k+j|k) + L_4 + M \; b_{pq\;4}(k+j|k)$$

$$\sum_{m=1}^{4} b_{pq\;m}(k+j|k) \leq 3$$

$$s_p(k+j+1|k) + t_\text{h} v_p(k+j+1|k) \leq L_1 + M \; b_{pq\;1}(k+j|k)$$

$$s_q(k+j+1|k) + t_\text{h} v_q(k+j+1|k) \leq L_2 + M \; b_{pq\;2}(k+j|k)$$

$$s_p(k+j+1|k) + t_\text{h} v_p(k+j+1|k) \leq s_q(k+j|k) + L_3 + M \; b_{pq\;3}(k+j|k)$$

$$s_q(k+j+1|k) + t_\text{h} v_q(k+j+1|k) \leq s_p(k+j|k) + L_4 + M \; b_{pq\;4}(k+j|k).$$

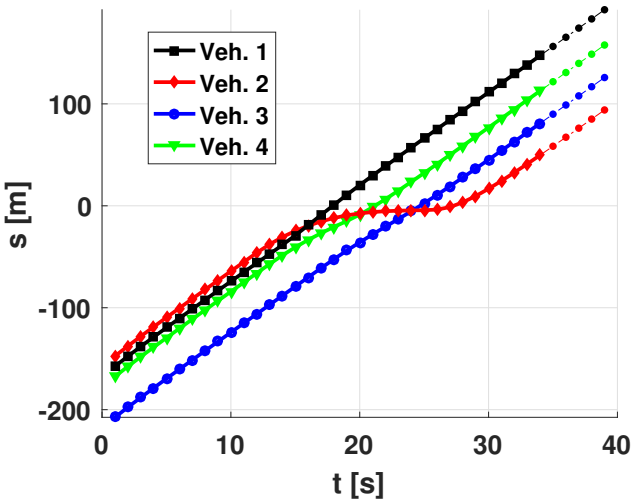


Figure 2.20: Positions shown for four merging vehicles with vehicle trajectories as solid lines and predictions as dashed lines and circles; $s_{\text{off}n=1...4}$=0, δt=1 s, t_{h}=2.1 s, N_{p} = 5, $w_{n=1...4}$=[0.25, 0.05, 0.1, 0.6], Q=1, R≃5.1. [6]

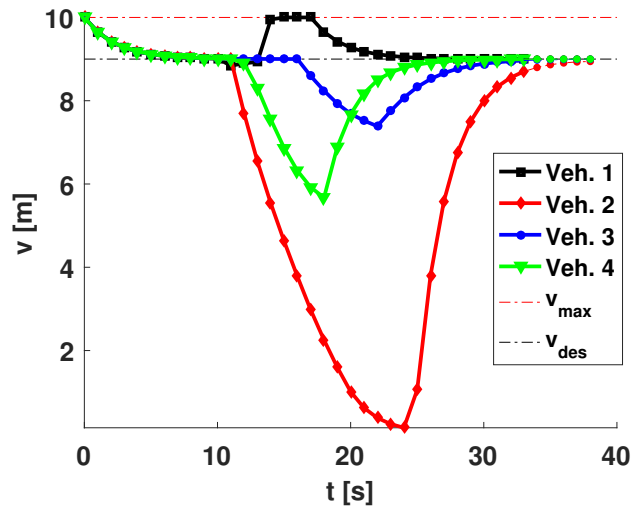


Figure 2.21: Speeds for four merging vehicles with vehicle speed values as solid lines and predictions as dashed lines and circles, δt = 1 s, t_{h} = 2.1 s, N_{p} = 5, $w_{n=1...4}$ = [0.25, 0.05, 0.1, 0.6], Q=1, R≃5.1. [6]

2.8 Computational speed and complexity

The simulations in Section 2.6 and Section 2.7 were run on a PC with Intel i7-4790 CPU and 16 GB memory. The Mixed-Integer Quadratic Program (MIQP) was solved using Gurobi v7.5.1 [32] via its Matlab interface. The computation times for a set of simulations computed for the state trajectories shown in Figure 2.14 are shown in Figure 2.22 **(a)**. The MIQP is solved efficiently for the non-conflicting part of the tests and for the car-following phase because there are relatively few nodes to be explored in the problem. However, there is a clear increase in computation time for decision-making around the merge, where many different binary values must be explored to obtain the global optimum. For two vehicles with time horizon length $N_\mathrm{p} = 25$ steps, which is $N_\mathrm{p}\delta t = 5$ s long, solution times were below the control time period. The computation times measured in the four-vehicle example in Section 2.7 are shown in Figure 2.22 **(b)**, which shows the same 5 s-long horizon but with $N_\mathrm{p} = 5$ steps. It should be noted that more efficient solution schemes can be obtained for the MIQP by exploiting structure. In [3], this was achieved through indicator binaries for collision-set avoidance. This work, does not attempt to optimise the method in this chapter, rather, it shows proof of safe time-headway formulation. Chapter 5 will inspect some of the potential performance enhancement options to improve the MIQP framework.

In conclusion, the computationally intensive cases are the problems that include non-trivial binary decisions; complexity peaks in the junctions, where the binary decisions are directly responsible for vehicle arbitration choices. In these cases, multiple convex sub-problems must be solved for a number of binary configurations before the branch-and-bound algorithm within the solver can return the globally optimal solution. For this reason, the binary decision variables are used sparingly to avoid a combinatoric explosion in complexity, which could render the problem impractical for real-time control.

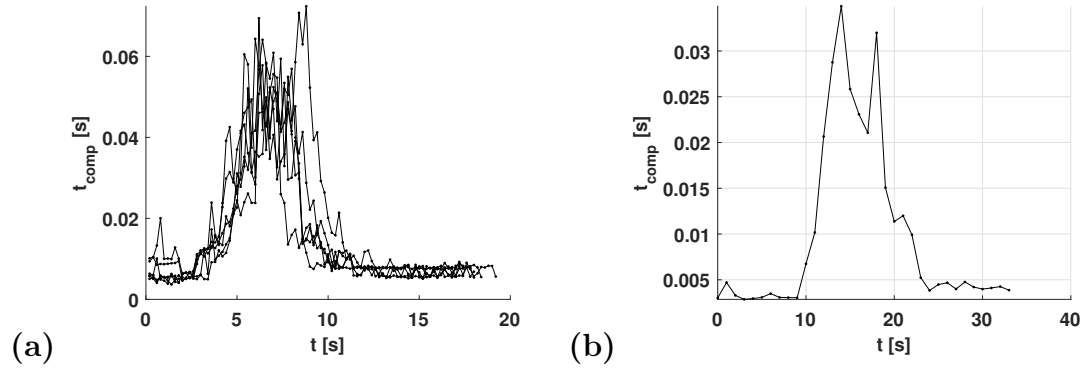


Figure 2.22: **(a)** Computation times for the Figure 2.14 simulations, δt = 0.2 s, $t_\mathrm{h} = 2.1$ s, $N_\mathrm{p} = 25$ simulations; **(b)** Computation times for Section 2.7 simulation (four vehicle merging) δt = 1 s, $t_\mathrm{h} = 2.1$ s, $N_\mathrm{p} = 5$, $w_{n=1\ldots4} = [0.25, 0.05, 0.1, 0.6]$. [6]

3

Cost and predictions

This chapter starts with a brief general overview of the cost function choices. Later in this chapter, the basic structure and elements of the cost functions used in this work are introduced in order to compare two formulations. The 'running-cost' formulation is widely used in the literature on vehicle control, but has unreachable speed reference when the vehicle is close to an obstacle or needs to stop, which is an operational mode for safety (see Chapter 2). This makes it difficult to obtain stability results; thus, stability results are acquired through a different cost formulation, with an objective corresponding to a soft form of the safety constraint. With proper parameter tuning (i.e. exact penalty), the same behaviours are observed with both cost formulations.

3.1 Overview

The velocity optimisation introduced in Chapter 2 uses binary variables to relax hyperplane constraints from obstacle avoidance in the hybrid-MPC formulation. According to *Theorem 1*, simple time-headway parameters provide positive-control-invariant sets to operate in and yield safe/recursively feasible operations. This chapter focuses on other aspects of the controller, namely the cost function that aims to encapsulate the goals and objectives in the optimised performance metric, while providing some stability results. For a more extensive introduction to the basics of cost functions, see [11, 59].

The approaches proposed in this work aim to obtain the globally optimal solution subject to obstacle avoidance. Obstacle avoidance is inherently an $\mathcal{NP}$-hard problem and in this work formulated into an $\mathcal{NP}$-complete problem with the use of binary relaxations. This enables commercially available branch-and-bound solvers to obtain the globally optimal solution, where, for each fixed binary configuration, the problem reverts itself to a convex QP optimisation.

It may be argued that formulating the problem to be a global method (to obtain globally optimal results) is very restrictive. For this reason, it may be better to solve

a less idealised formulation to sub-optimality with local solvers than to get the globally optimal solution to a cruder problem.

It should be noted that when problem convexity is less of a concern, more options are available for the problem formulations. These include non-convex cost functions (e.g. general utility functions) [83], non-linear system dynamics (e.g. steering in 2D problems) [67] and non-convex constraints in general (e.g. non-convex vehicle shapes and obstacles) [68].

Economic MPC is concerned with optimising generally non-convex cost functions. For example, non-convex utility functions can include fuel economy, gear selection and transmission costs, regenerative braking, battery management, and minimal time objectives.

It is generally recognised in the literature that using convex QP cost function from linear control theory is inferior compared to the freedom to describe more general objectives with utility functions available in economic MPC.

This work does not aim to settle on a single non-linear utility function, note that this does not prevent the framework to be considered for such cases. The choice made here is to facilitate the global method in a simple, low complexity way which allows global solvers to compute the globally optimal solutions. Solver choices for the QPs are numerous with mature techniques yielding stable outputs, which is valuable if optimal control is used for safety critical applications. State-of-the-art commercial solvers are currently able to handle MIQPs at unprecedented speeds, which are expected to continue improving.

3.2 Cost inspection

The so-called 'running-cost' formulation, which is a common baseline approach in the literature on vehicle control, was implemented in the MIQP examples in Chapter 2.

$$J = q\,\left(v\left(k+N_{\mathrm{p}}|k\right)-v_d\right)^2+\sum_{j=0}^{N_{\mathrm{p}}-1} q\,\left(v\left(k+j|k\right)-v_d\right)^2+r\;u\left(k+j|k\right)^2, \tag{3.1}$$

where the desired speed or setpoint is v_d. Cost functions may aggregate several cost and penalty terms if the intended applications and objectives require their use. Commonly used cost penalties are imposed on decision variables: states in vector x, control inputs u, and control input change Δu (total variance); additional penalties can come from soft constraint violation, and logic-imposed penalties. A constant shift term may be present in the cost functions and it may be omitted from the mathematical optimisation without changing the optimal solution of decision variables [11]. For engineers, the exact cost with its weight and scale usually bear less significance, aside from those that affect solver performance. In contrast, cost in other fields, such as finance, may bear a specific monetary unit (dimension), meaning an optimal cost value could be important in further calculations or presentations. Engineers, however, generally design controllers to actively

keep reasonably small state errors, control actions, and minimal soft-constraint violations, while providing stable and robust operations. Cost only bears some informative value; its trends are typically used to prove stability.

Convex problems are easy to solve; one requirement for a convex problem is the convex cost function. Convex cost functions may be formed in terms of l_1, l_∞ norms and/or weighted-l_2 norms that can be cast as the sum of weighted squares. A mixed-integer program (MIP) is considered non-convex, however, when the integer variables are fixed to a configuration $b = \hat{b}$; the optimisation task can revert back to convex if the cost function and constraints are all convex [11]. The constant offset arising from model mismatch and uncertainties may simply be handled by introducing an integrator in the form of additional input change variables Δu and incorporating the control inputs u as additional states in the system dynamics [11].

Uncertainties, model errors, offsets and constraint violations are not the main concerns of this work, which assumes an ideal scenario. When the types of additional error models to be handled are identified in a clear application (e.g. real vehicle tests), then these models, constraints and cost penalties can be added via robust optimal control techniques, such as tightening and relaxing constraints with slack variables.

The parts of a finite horizon cost function can be classified on the type of decision variables they defined over, such as Mayer, Lagrange, and, Bolza, the last of which is a combination of the former two types. The Mayer cost is a function of final states; the Lagrange is the integral of the combination of state and control input costs over the finite horizon (except the final states). Transformation between cost types can be done through additional states, integration, or derivation [38]. It is particularly important that, in an ideal case (e.g. in dynamic programs), the cost spans over infinity, which requires calculations with an infinite time horizon. Infinite-time predictions for discrete-time systems are modelled as a Markov decision process (MDP) [8], which has $\mathcal{NP}$-complete complexity. In practice, calculating the infinite-horizon cost is computationally expensive, intractable, and sometimes impossible. In the face of inherent uncertainties associated with real processes, the impact of the far future diminishes while the more immediate control actions become more important. Treating these issues with MPC is a practical solution:

$$J = \phi_{\text{fh}} + \hat{\phi}_{\text{ih}}, \tag{3.2}$$

where the cost component over the finite horizon is $\phi_{\text{fh}}(t)$, $t = \left[t_k, t_{k+N_\text{p}}\right]$ with a high resolution and some approximation of the rest of the infinite horizon cost-to-go is $\hat{\phi}_{\text{ih}}(t)$, $t = \left(t_{k+N_\text{p}},\ \infty\right]$.

The Bolza cost, which incorporates both the Mayer terminal cost and the Lagrange cost, is dependent on the inner-horizon states and controls and terminal states. The continuous-time integral in the Lagrange cost is approximated as a finite summation in

discrete-time control:

$$J = \underbrace{\sum_{j=0}^{N_\mathrm{p}-1} l(x(k+j|k), u(k+j|k))}_{\text{Lagrange cost}} + \underbrace{F(x(k+N_\mathrm{p}|k))}_{\text{Terminal cost}}, \tag{3.3}$$

where the stage costs are $l(x(k+j|k), u(k+j|k))$, which correspond to discrete-time states and controls. In the case of the 'running-cost' formulation in (3.1), the stage cost was:

$$l(v(k+j|k), u(k+j|k)) = q\big(v(k+j|k) - v_d\big)^2 + ru(k+j|k)^2, \tag{3.4}$$

where the terminal cost was defined, as shown: $F(x(k+N_\mathrm{p}|k)) = q\big(v(k+N_\mathrm{p}|k) - v_d\big)^2$.

3.3 Terminal-position-based cost function

This section introduces the terminal-position-based or l_1-norm-based cost, which is further analysed in this chapter and shown to be related to the 'running cost' under certain conditions. The aim was to propose a cost function that has zero reference speed in the formulation; otherwise, a stationary vehicle near an obstacle would have unreachable setpoint due to the shape of the Ω invariant set, which would be in conflict with the requirements for Lyapunov stability [53]. Thus, the chosen cost form is:

$$\begin{aligned} J = \sum_{j=0}^{N_\mathrm{p}-1} &\left(qv(k+j|k)^2 + ru(k+j|k)^2\right) + q_\mathrm{f} v(k+N_\mathrm{p}|k)^2 \\ &+ \rho\left|s(k+N_\mathrm{p}|k) + \beta v(k+N_\mathrm{p}|k) - s_G\right|, \end{aligned} \tag{3.5}$$

where the weighting factors are q, r, q_f, β and ρ, and the goal position is s_G. The Mayer cost in this case has a squared terminal speed term, similarly to the stage costs and an additional l_1-norm. The l_1-norm part incorporates a similar expression encountered during simple time-headway safety derivation in *Theorem 1.* The total cost at the goal position assuming stationary vehicle gives zero, required for the Lyapunov stability analysis. In the original 'running-cost' formulation, the quadratic-cost function was based only on the speed states and control inputs; the new formulation includes the terminal position and speed in the cost. Equilibrium system states are identified using the state evolution from (2.8):

$$\begin{bmatrix} s_{eq} \\ v_{eq} \end{bmatrix} = A \begin{bmatrix} s_{eq} \\ v_{eq} \end{bmatrix} + Bu_{eq},$$

where equilibrium states are arbitrary in means of the position s_{eq}, while the equilibrium speed is $v_{eq} = 0$ and the control input is assumed to be zero ($u_{eq} = 0$). When the

equilibrium coincides with the goal position for all states (at $s_{eq} = s_G$, $v_{eq} = 0$, $u_{eq} = 0$) then the cost from (3.5) gives $J = 0$, otherwise, the cost is $J > 0$ required for Lyapunov stability condition [53, 59].

In a simple test scenario a vehicle is approaching a goal position using MPC 3.1; when experimentally tuning the cost function parameter choices, it can be noted that the vehicle speed converges to a settable constant value far from the goal position. Furthermore, the optimal vehicle speed profile has a gradual deceleration phase close or past the goal position.

MPC-3.1

$$
\begin{aligned}
&J^*_{\text{MPC-3.1}} = \min\ (3.5) \\
&\text{s.t.} \quad \forall j \in \{0 \dots (N_\text{p} - 1)\} : \\
&\qquad\qquad x\,(k|k) = x\,(k) \\
&\qquad x\,(k+j+1|k) = A\ x\,(k+j|k) + B\ u\,(k+j|k) \\
&\qquad\qquad (x, u) \in \mathcal{X} \times \mathcal{U},
\end{aligned}
$$

where as a reminder the state and control sets are $\mathcal{X} = \{s, v \mid v_\text{min} \leq v \leq v_\text{max}\}$ and $\mathcal{U} = \{u \mid a_\text{min} \leq u \leq a_\text{max}\}$, respectively.

For further analysis, the l_1-norm part of the cost function is inspected by branching, which provides three cases:

$$
|s + \beta v - s_G| = \begin{cases} -\left(s + \beta v - s_G\right), & s + \beta v - s_G < 0 \\ 0, & s + \beta v - s_G \equiv 0 \\ s + \beta v - s_G, & s + \beta v - s_G > 0 \end{cases} \tag{3.7}
$$

3.3.1 Branching case 1

For case 1, the region of inspection is $s(k + N_\text{p}|k) + \beta v(k + N_\text{p}|k) - s_G < 0$, and the cost function takes the form:

$$
\begin{aligned}
J = &\sum_{j=0}^{N_\text{p}-1} \left(q v(k+j|k)^2 + r u(k+j|k)^2\right) + q_\text{f} v(k+N_\text{p}|k)^2 \\
&- \rho(s(k+N_\text{p}|k) + \beta v(k+N_\text{p}|k) - s_G).
\end{aligned} \tag{3.8}
$$

Using the state calculation of LTI dynamics (2.8):

$$
x(k+N_\text{p}) = A^{N_\text{p}} x(k) + \sum_{i=0}^{N_\text{p}-1} A^i B u(k+N_\text{p}-1-i) \tag{3.9}
$$

and

$$u(k) = \frac{v(k+1) - v(k)}{\delta t} \tag{3.10}$$

gives the terminal position in terms of:

$$s(k+N_\mathrm{p}) = s(k) + \big(\frac{1}{2}v(k) + v(k+1) + v(k+2) + \ldots + v(k+N_\mathrm{p}-1) + \frac{1}{2}v(k+N_\mathrm{p})\big)\delta t. \tag{3.11}$$

Now, (3.11) is substituted back to (3.8) and yields:

$$\begin{aligned} J = &\sum_{j=0}^{N_\mathrm{p}-1} \big(qv(k+j|k)^2 - \rho\delta t v(k+j|k) + ru(k+j|k)^2\big) + q_\mathrm{f} v(k+N_\mathrm{p}|k)^2 \\ &- \rho(\frac{\delta t}{2} + \beta)v(k+N_\mathrm{p}|k) - \rho(s(k|k) - s_G) + \rho\frac{\delta t}{2}v(k|k), \end{aligned} \tag{3.12}$$

where initial states are constants $s(k|k) = s(k)$ and $v(k|k) = v(k)$, which are to be collected in the constant cost shift $\hat{J}_{const1}$.

$$\begin{aligned} J = &\sum_{j=0}^{N_\mathrm{p}-1} \big(qv(k+j|k)^2 - \rho\delta t v(k+j|k) + ru(k+j|k)^2\big) \\ &+ q_\mathrm{f} v(k+N_\mathrm{p}|k)^2 - \rho(\frac{\delta t}{2} + \beta)v(k+N_\mathrm{p}|k) + \hat{J}_{const1} \end{aligned} \tag{3.13}$$

which alternatively can be expressed as:

$$J_1 = \sum_{j=0}^{N_\mathrm{p}-1} \Big(q\big(v(k+j|k) - v_d\big)^2 + ru(k+j|k)^2\Big) + q_\mathrm{f}\big(v\,(k+N_\mathrm{p}|k) - v_{df}\big)^2 + J_{const1}, \tag{3.14}$$

where the desired speed is $v_d = \frac{\rho\delta t}{2q}$ and the terminal desired speed is $v_{df} = \frac{\rho(\frac{\delta t}{2}+\beta)}{2q_\mathrm{f}}$ for $q,\ q_\mathrm{f} > 0$. This form is closely related to the 'running-cost' shifted by a constant and, thus, belongs to the family of reference-tracking MPC, which minimises the state error $\xi = v - v_\mathrm{ref}$, where the setpoint is v_ref. By choosing $q_\mathrm{f} \equiv q$, a connection between (3.13) and (3.1) is evident, wherein the only difference is a constant shift (the constant shift is indifferent in means of the optimal solutions returned by optimisation problems, as argued before).

Thus, in summary, if the optimisation in branching case 1 would be formulated as a separate optimisation it would take the form of MPC 3.2.

MPC-3.2

$$
\begin{aligned}
&J^*_{\text{MPC-3.2}} = \min\ (3.8)\\
&\text{s.t.}\quad \forall j \in \{0 \dots (N_\text{p}-1)\}:\\
&\qquad\qquad x\,(k|k) = x\,(k)\\
&\qquad x\,(k+j+1|k) = A\,x\,(k+j|k) + B\,u\,(k+j|k)\\
&\qquad s(k+N_\text{p}|k) + \beta v(k+N_\text{p}|k) - s_G < 0 \qquad (3.15a)\\
&\qquad\qquad (x,u) \in \mathcal{X} \times \mathcal{U},
\end{aligned}
$$

This is generally the active and desired case when the controller parameters are appropriately tuned. In the original case in MPC 3.1 with (3.5) cost, the terminal speed and position adapts and avoids the region $s(k+N_\text{p}|k) + \beta v(k+N_\text{p}|k) \geq s_G$ as it were an obstacle. Now, in MPC 3.2 notice that the valid domain is appearing as a hard constraint (3.15a) in the optimisation, just like an obstacle avoidance constraint would. In the case of MPC 3.1 overrunning is possible due to the softness of the objective, however, for branching case 1, this is by definition an invalid region, when terminal states violate the slope constraint (3.15a).

3.3.2 Branching case 2

Case 2 occurs when $s(k+N_\text{p}|k) + \beta v(k+N_\text{p}|k) - s_G \equiv 0$, meaning that the setpoint for speed jumps to zero for both mid- and terminal-speed states. In this case, the setpoint (v_ref) is zero and the terminal state slides along the $s(k+N_\text{p}|k) + \beta v(k+N_\text{p}|k) = s_G$ line, shown in Figure 3.1. When zero is reached for all v states and u controls, the system is in equilibrium ($J_{const2} = 0$).

$$J_2 = \sum_{j=0}^{N_\text{p}-1} \left(q v(k+j|k)^2 + r u(k+j|k)^2\right) + q_\text{f} v(k+N_\text{p}|k)^2, \qquad (3.16)$$

3.3.3 Branching case 3

Case 3 occurs when $s(k+N_\text{p}|k) + \beta v(k+N_\text{p}|k) - s_G > 0$

$$J_3 = \sum_{j=0}^{N_\text{p}-1} \left(q(v(k+j|k)+v_d)^2 + r u(k+j|k)^2\right) + q_\text{f}(v(k+N_\text{p}|k)+v_{df})^2 + \tilde{J}_{const3} \qquad (3.17)$$

Minimising this cost (minimise J_3), which is subjected to non-negative speeds, would result in a fast deceleration phase. In this case, the speed error is referenced around the negative speed (the speed error is $\xi = v - v_\text{ref} = v + v_d$, thus $v_\text{ref} = -v_d$); this is a completely

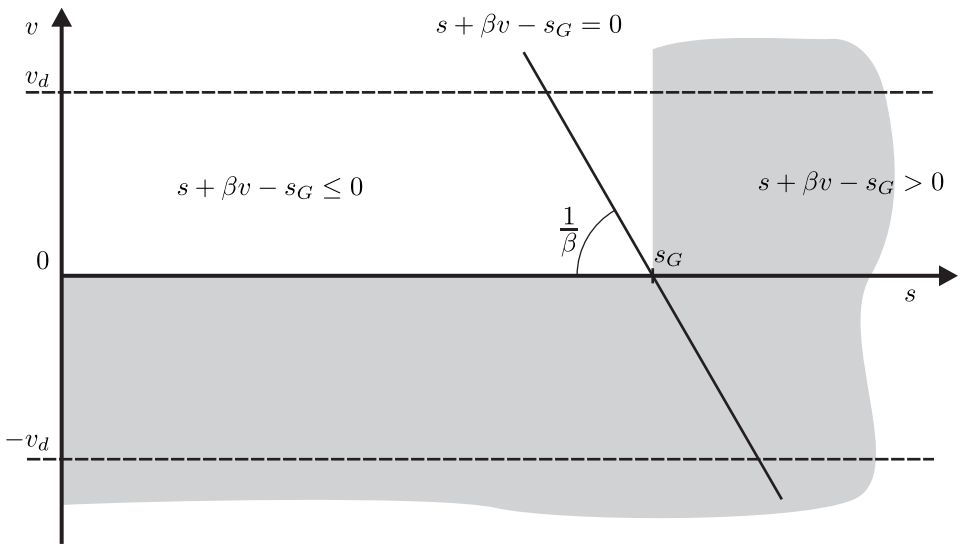


Figure 3.1: The setpoint jumps between v_d, 0 and $-v_d$ in line with the transition of $s(k + N_\mathrm{p}|k) + \beta v(k + N_\mathrm{p}|k) - s_G$ from a negative to a positive value.

unreachable setpoint, as it is in conflict with $v \geq 0$ constraint ($v_\mathrm{ref} = -v_d \leq 0 \leq v$). Figure 3.1 shows the regions of respected cases of the l_1-norm and the possible reference speeds. It is forbidden and undesirable to have states of negative speeds or positions passing through the goal corresponding to the grey areas in Figure 3.1. Switching between the three possible cost cases opens the way to simplifying and modelling them as LQR controls with the choice to find stabilising tuning parameters. This makes it possible to form the appropriately tuned operating case, which results in a mathematical optimisation identical to that of the 'running-cost' formulation with added terminal set constraints.

3.3.4 Tuning the controller

At this point, there are several parameters in the controller: q, r, q_f, ρ and β. Furthermore, as previously identified, desired speed v_d can be encompassed in the formulation that appears as a reference speed in cases decided by an inherent switching of the terminal cost. The method for tuning entails choosing the stabilising linear control for the quadratic cost function. The three cases just discussed are reduced to LQR forms dependent on q, r and q_f parameters, generally minimising the speed-state errors $\xi = v - v_\mathrm{ref}$ to the reference speed v_ref. For simplicity, let the setpoint v_ref be the same for each time step, making the general LQR form:

$$J = \sum_{j=0}^{N_\mathrm{p}-1} \left(q\ \xi(k+j|k)^2 + ru(k+j|k)^2 \right) + q_\mathrm{f}\ \xi(k+N_\mathrm{p}|k)^2. \tag{3.18}$$

Assume the following closed-loop control law:

$$u = -Kv \tag{3.19}$$

where the closed loop gain is K and not yet determined. When the terminal state reaches the desired surface defined by $s + \beta v - s_G = 0$, it will ideally follow this surface until reaching the equilibrium state at s_G; this happens in case 2, where the l_1 norm remains at zero. Thus, similarly to the positive invariance calculations, let the states coincide with $s + \beta v - s_G = 0$ line in two consecutive time steps:

$$s_k + \beta v_k - s_G = s_{k+1} + \beta v_{k+1} - s_G,$$

with substitution of:

$$s_{k+1} = s_k + v_k \delta t + a_k \frac{\delta t^2}{2}$$
$$v_{k+1} = v_k + a_k \delta t$$

gives

$$v_k \delta t + a_k \frac{\delta t^2}{2} + \beta a_k \delta t = 0$$

$$a_k = -\frac{1}{\frac{\delta t}{2} + \beta} v_k.$$

This defines the gain from (3.19):

$$K = \frac{1}{\frac{\delta t}{2} + \beta} \tag{3.20}$$

which can be used to calculate the q, r and q_f parameters.

The Discrete Algebraic Ricatti Equation (DARE) is:

$$A^T Q_\mathrm{f} A - Q_\mathrm{f} - A^T Q_\mathrm{f} B \left(B^T Q_\mathrm{f} B + R\right)^{-1} B^T Q_\mathrm{f} A + Q = 0,$$

where the positive LQR cost matrices are Q, R and Q_f—which, in this formulation, are only one-dimensional (scalars)—and, respectively, the parameters q, r and q_f. Furthermore, the optimal closed-loop gain is $K = \left(B^T Q_\mathrm{f} B + R\right)^{-1} B^T Q_\mathrm{f} A$. The velocity optimisation may be formulated exclusively through the use of speed states and control inputs, as the positions are not utilised in the current unconstrained tuning; thus, the state matrices are $A = 1$ and $B = \delta t$. The DARE simplifies to:

$$q = A^T q_\mathrm{f} B \; K = q_\mathrm{f} \; K \; \delta t,$$

and the gain equation to

$$K = \frac{q_\mathrm{f} \delta t}{q_\mathrm{f} \; \delta t^2 + r}.$$

Using the previously required gain setting to remain on the $s + \beta v - s_G = 0$ line:

$$q_{\mathrm{f}} = \frac{r}{\left(\frac{1}{K} - \delta t\right)\delta t} = \frac{r}{\left(\beta - \frac{\delta t}{2}\right)\delta t}.$$

Furthermore, the scale of J in the LQR is arbitrary and can be chosen by fixing one of the three parameters (q, r, or q_{f}). For simplicity, let:

$$\begin{aligned} r &\equiv 1, \\ q_{\mathrm{f}} &= \frac{1}{\left(\beta - \frac{\delta t}{2}\right)\delta t} \\ q &= \frac{1}{\left(\beta - \frac{\delta t}{2}\right)\left(\beta + \frac{\delta t}{2}\right)} = \frac{1}{\beta^2 - \frac{\delta t^2}{4}}. \end{aligned} \tag{3.21}$$

To keep the parameters positive, $\beta - \frac{\delta t}{2} \geq 0$; thus, similarly to the requirements in *Theorem 1*, it is true that $0 \leq \delta t \leq 2\beta$.

3.3.5 Stability

The three cases outlined above can be reduced to LQR controllers through which the reference speed is chosen by the terminal speed state and position prediction:

$$v_{\mathrm{ref}} = \begin{cases} v_d, & s(k+N_{\mathrm{p}}|k) + \beta v(k+N_{\mathrm{p}}|k) - s_G < 0, \quad \text{(Case 1)} \\ 0, & s(k+N_{\mathrm{p}}|k) + \beta v(k+N_{\mathrm{p}}|k) - s_G \equiv 0, \quad \text{(Case 2)} \\ -v_d, & s(k+N_{\mathrm{p}}|k) + \beta v(k+N_{\mathrm{p}}|k) - s_G > 0, \quad \text{(Case 3)} \end{cases} \tag{3.22}$$

In case 3, the reference speed is unreachable because only positive speeds are allowed (i.e. in this framework, the vehicle cannot go to its goal position in reverse if it was missed). Case 2 has the stopping velocity profile with immediate deceleration due to the tuned LQR parameters, converging to an equilibrium state corresponding to $v_{k\to\infty} = 0$, $u_{k\to\infty} = 0$. However, if $s(k+N_{\mathrm{p}}|k) \leq s_G$ and $v(k+N_{\mathrm{p}}|k) = 0$ then case 1 will be switched to where $v_{\mathrm{ref}} = v_d > 0$; thus, the vehicle will be in motion at some point in the horizon. If the terminal position $s(k+N_{\mathrm{p}}|k) = s_G$ or the terminal states are on the line of $s(k+N_{\mathrm{p}}|k) + \beta v(k+N_{\mathrm{p}}|k) - s_G \equiv 0$ then (since $r > 0$), the homogeneous penalty over all control inputs in the horizon ensures that the acceleration to reach the goal is spread across the time horizon. This is important because doing so results in some degree of initial acceleration $v(k+1|k) > 0$, meaning that no idling or loitering happens in the finite horizon (otherwise, $v = 0$ sections could form). Furthermore, the vehicle would build up speed even if it had been stationary (i.e. $v(k) = 0$), for example, after a full stop before a junction. Finally, $v(k+1|k) > 0$ ensures that $s(k+1)$ increases over $s(k)$ when $s(k) + \beta v(k) < s_G$ (i.e. the vehicle starts moving towards the goal position, ergo the equilibrium state).

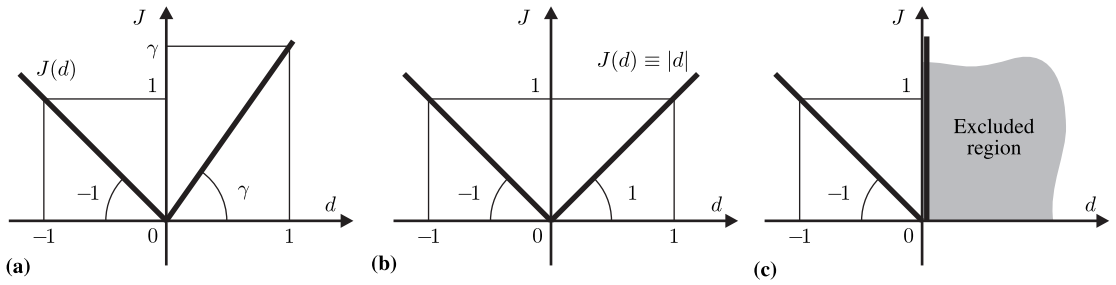


Figure 3.2: (a) The general linear non-negative penalty function that has a tunable right slope with γ parameter; **(b)** the l_1 norm can be formulated from (a) with $\gamma = 1$; **(c)** a penalty with a linear-left and hard-right side, where $\gamma = \infty$.

Acceleration limits from (2.11) and the control law (3.19) constitute other design considerations; these would impose limits on K, the closed-loop gain should be chosen:

$$a_{\min} \leq -K\xi \leq a_{\max}, \tag{3.23}$$

where $\xi = v - v_{\text{ref}}$ is the speed error.

3.3.6 Soft constraint transformation

As for when the control parameters are not appropriately tuned and the linear control law would demand higher deceleration to follow the $s(k + N_{\text{p}}|k) + \beta v(k + N_{\text{p}}|k) = s_G$ line, the trajectory can overrun this line. This is potentially undesirable since, if the goal position cannot be relaxed (e.g. end of a parking space), the vehicle may run out of road and hit an obstacle.

Through the introduction of an additional slack variable and constraints to the optimisation, they can be used to create a settable linear penalty function, where the slope of the right side can be changed through a parameter:

$$\begin{aligned} &J(d) = \min_{\varepsilon} -d + \varepsilon \\ &s.t.: \\ &(1+\gamma)d - \varepsilon \leq 0 \\ &\qquad\qquad \varepsilon \geq 0, \end{aligned} \tag{3.24}$$

where the input argument is d, the slack variable is ε and the slope of the penalty cost is set by γ. Figure 3.2 **(a)** shows the resultant cost dependence on the input variable.

Notice that the l_1 norm can be decomposed in the previous form when $\gamma = 1$:

$$
\begin{aligned}
&|d| = J(d) = \min_{\varepsilon} -d + \varepsilon \\
&s.t.: \\
&2d - \varepsilon \leq 0 \\
&\qquad \varepsilon \geq 0
\end{aligned}
$$

which is shown in Figure 3.2 **(b)**.

Recall that the original cost analysed in (3.5) sets the reference speed (3.22) in case 3 to negative for the LQR form (i.e. $v_{\text{ref}} = -v_d = -\frac{\rho \delta t}{2q}$). However, this can be arbitrarily changed when the generalised soft-penalty formulation from (3.24) is being implemented with the γ parameter. Softened constraints are beneficial because their violation is feasible; thus, the controller may operate even after, for example, inadvertently passing the goal position. It is possible to choose the parameter tuning with l_1 and l_∞ norm soft penalties to give back the feasible hard-constraint-equivalent optimal trajectories using the exact penalty method [41] or approximate l_2 penalty function cases. In this case, the design considerations allow the controller to operate without entering case 3, which otherwise would necessitate the soft penalty formulation.

However, this work does not focus on softening obstacle avoidance constraints; by choosing the soft-constraint penalty γ to ∞, or fixing $\varepsilon = 0$ and eliminating the softening, case 3 is made a forbidden region. A hard-constraint case is shown in Figure 3.2 **(c)**. This keeps the operating modes in cases 1 and 2 attainable but requires positive control-invariant consideration for the terminal set to remain recursively feasible. Thus, it is the additional hard constraint that retains the form:

$$d \leq 0.$$

By substitution of the original l_1-norm argument in place of d gives:

$$s(k + N_{\text{p}}|k) + \beta v(k + N_{\text{p}}|k) - s_G \leq 0$$

hard constraint and

$$
\begin{aligned}
&\beta - \frac{\delta t}{2} \geq 0, \\
&\beta + \frac{\delta t}{2} = \frac{1}{K} \leq -\frac{v_{\max}}{a_{\min}}
\end{aligned}
$$

requirements.

3.3.7 Numerical examples

This section highlights the properties of the controllers discussed previously. The two MPC formulations are the soft formulation, MPC 3.3, with the terminal-position-based cost, and the hard formulation, MPC 3.4, with the 'running-cost' and hard constraint condition. When tuned appropriately the two MPCs are expected to generate identical trajectories for unconstrained cases (disabled speed and acceleration limits). Moreover, in constrained cases where the test problems are set up to induce constraint violations, the soft formulation is expected to remain feasible, in contrast to the hard 'running-cost' formulation, which is expected to become infeasible on hard constrain violation.

The soft formulation, MPC 3.3, is defined without the state (speed) limits (3.25b) and control input limits (3.25c). MPC 3.3 is:

MPC-3.3, MPC formulation with l_1-norm based terminal-cost

$$J^*_{\text{MPC-3.3}} = \min \sum_{j=0}^{N_\text{p}-1} \left(qv(k+j|k)^2 + ru(k+j|k)^2 \right) + q_\text{f} v(k+N_\text{p}|k)^2$$
$$+ \rho \, |s(k+N_\text{p}|k) + \beta v(k+N_\text{p}|k) - s_G| \tag{3.25a}$$

s.t. $\forall j \in \{0 \dots (N_\text{p}-1)\}$:

$$x\,(k|k) = \; x\,(k)$$
$$x\,(k+j+1|k) = \; A\; x\,(k+j|k) + B\; u\,(k+j|k)$$
$$v(k+j+1|k) \geq 0$$
$$\cancel{x \in \mathcal{X}} \tag{3.25b}$$
$$\cancel{u \in \mathcal{U}} \tag{3.25c}$$

The hard 'running-cost' formulation is MPC 3.4 subject to only the equalities of the dynamics and the hard constraint (3.26d) with similarly disabled speed limits (3.26b) and control input limits (3.26c):

MPC-3.4, MPC formulation with 'running-cost' and hard constraint

$$J^*_{\text{MPC-3.4}} = \min \sum_{j=0}^{N_\text{p}-1} \left(q\xi(k+j|k)^2 + ru(k+j|k)^2\right) + q_\text{f}\xi(k+N_\text{p}|k)^2 \tag{3.26a}$$

s.t. $\forall j \in \{0 \dots (N_\text{p}-1)\}$:

$$x\,(k|k) = \; x\,(k)$$

$$x\,(k+j+1|k) = \; A\; x\,(k+j|k) + B\; u\,(k+j|k)$$

$$v(k+j+1|k) \geq 0$$

$$\cancel{x \in \mathcal{X}} \tag{3.26b}$$

$$\cancel{u \in \mathcal{U}} \tag{3.26c}$$

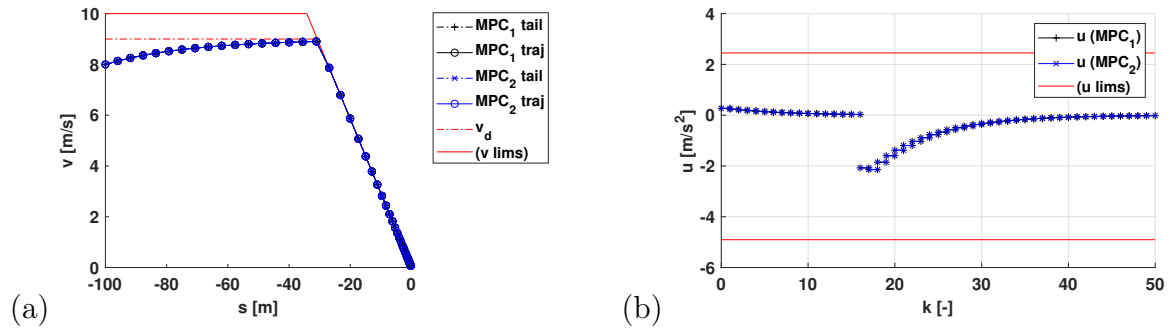
$$s(k+N_\text{p}|k) + \beta v(k+N_\text{p}|k) - s_G \leq 0, \tag{3.26d}$$

where the speed errors are $\xi(k+j|k) = v(k+j|k) - v_d$.

Note that the following experiments are only for demonstration purposes obtaining the unconstrained trajectories, as the formulations would naturally behave without the operational state and control limits defined, while initial states of the simulation are still chosen within the state limits to correspond with relevant cases.

The first simulation parameters are $\delta t = 0.5$ s, $N_\text{p} = 1$, $s_G = 0$ m, $v_d = 9$ ms^{-1} and $x(0) = [-100, 8]^T$ with initial position and speed in m and ms^{-1}, respectively. Parameters q, r, q_f and ρ were chosen with $\beta = 3.4197$ s from (3.21). Furthermore, the limits, if they would have been imposed, were $a_\text{max} = 9.81/4$ ms^{-2}, $a_\text{min} = -9.81/2$ ms^{-2} and $v_\text{max} = 10$ ms^{-1}. The modelling was done using CVX, a package for specifying and solving convex programs in Matlab [16, 28].

In Figure 3.3 **(a)**, the trajectories of the two MPCs can be seen coinciding as expected. In the first section, both controllers approach the v_d reference speed. Later, a slowdown period can be seen respecting the chosen $1/\beta$ slope on the v–s graph converging towards the goal position. The shown accelerations in Figure 3.3 **(b)** have monotonic decreasing trends after the initial tracking error and the switch to goal approaching phase. In Figure 3.4, the optimal cost of MPC 3.3 goes to zero (i.e. $J([s_G, 0]) \equiv 0$) in contrast with



(a) (b)

Figure 3.3: Comparison of the two MPC controller formulations: MPC 3.3 (MPC_1) and MPC 3.4 (MPC_2) **(a)** v–s and **(b)** u–k.

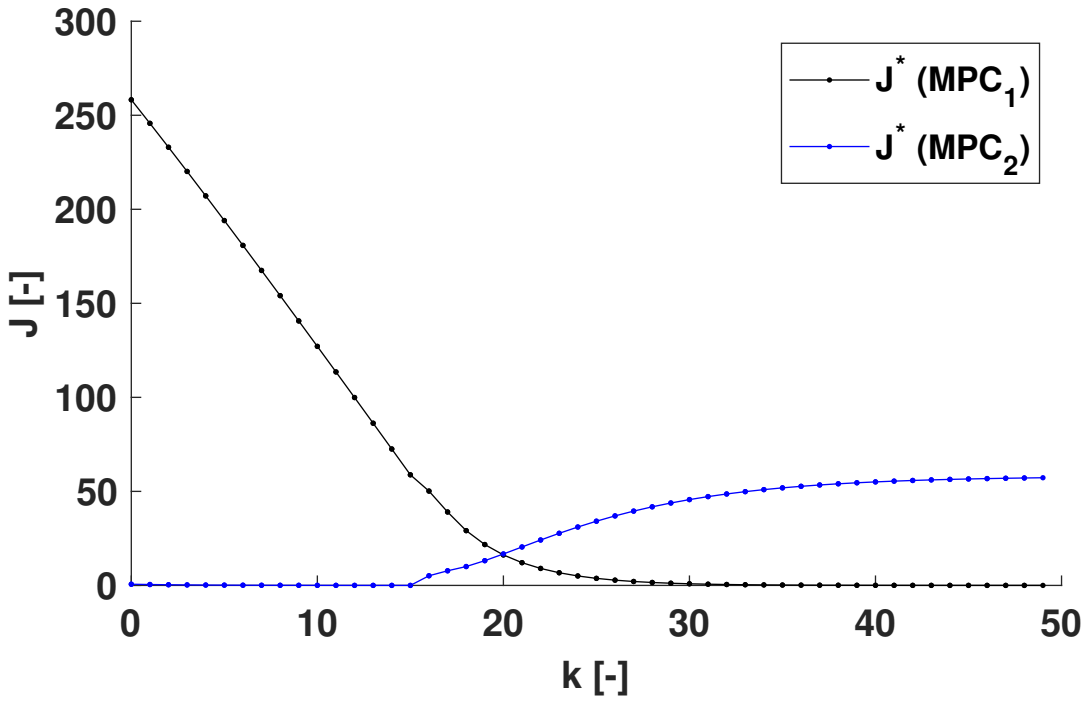


Figure 3.4: The optimal cost of MPC 3.3 (MPC_1) goes to zero in the goal position equilibrium; MPC 3.4 (MPC_2) 'running-cost' formulation maintains zero cost through ideal tracking of v_d and increases in cost as the speed decreases to zero.

MPC 3.4, which has zero cost for undisturbed settled tracking of v_d while in the goal approaching phase the speed setpoint becomes unreachable and the 'running-cost' tracking error rises to a constant value.

From the dual-mode structure of MPC 3.4, the terminal cost—the cost-to-go corresponding to the equivalent LQR—represents the error minimisation to v_d. The terminal states are constrained, however, this does not apply to the rest of the infinite horizon, as it would yield infinite cost-to-go. This can be seen in Figure 3.5, where the controller assumes an immediate speed-up following the terminal state. In order to respect acceleration limits, the LQR K (β) parameter should be chosen with the lower acceleration limit in mind:

$$\beta + \frac{\delta t}{2} = \frac{1}{K} \leq \min(-\frac{v_{\max}}{a_{\min}}, \frac{v_d}{a_{\max}}) \tag{3.27}$$

Figure 3.6 shows a well-designed case, as the acceleration respects the limits without explicitly imposing control input constraints in the problem, this means that the infinite horizon tail results in realistic behaviour.

Remark: since the vehicle acceleration and deceleration properties are different in most cases, the controller parameters are advised to respect the more constraining limit. In this situation, $v_d/a_{\max}$ is the lower value than $-v_{\max}/a_{\min}$. Note that $s(k+N_\text{p}|k) + \beta v(k+N_\text{p}|k) \leq s_G$ constraint is representing the right side of the Ω set in *Theorem 1*, thus choosing β below the critical t_h parameter preserving recursive feasibility or the same

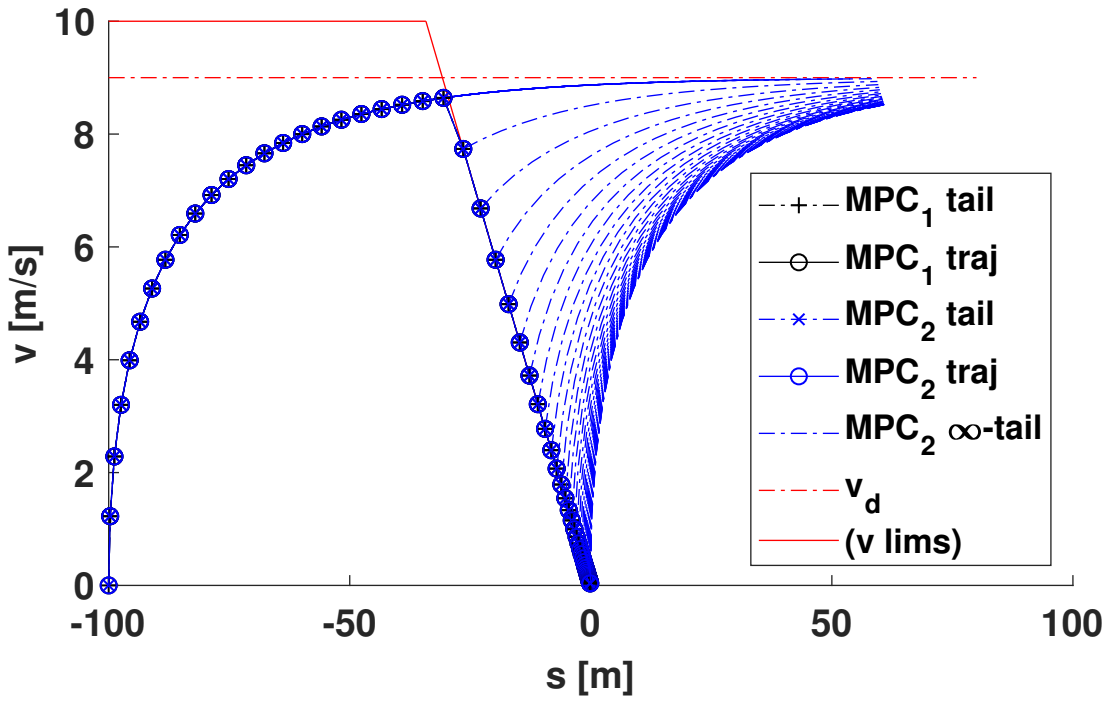


Figure 3.5: MPC 3.3 (MPC_1) and MPC 3.4 (MPC_2) trajectories with the rest of the dual-mode infinite horizon tail of MPC 3.3 partially indicated.

condition identified in Section 3.3.4 would lead to the loss of positive control invariance and recursive feasibility for the 'running-cost' formulation in MPC 3.4 when all state and control limits are enabled.

In the following, let the safety condition in the simulations be disregarded. Let $\beta = (-v_{\max}/a_{\min} - \frac{\delta t}{2}) * 0.6$ s, which is 60% of what would be needed for positive control invariance. Figure 3.7 shows the state and control input trajectories when the limit constraints are disabled. The theoretical acceleration limits are exceeded in both speeding-up and slowing-down phases, Figure 3.7 **(b)**, implies that catastrophic feasibility problems may arise in the hard-constrained case due to the peak deceleration in the slow-down phase.

In the next simulations, state and control limits are added to both formulations (i.e. constraints (3.25b), (3.25c), (3.26b) and (3.26c) are enabled), completing the terminal set; results are displayed in Figure 3.8 for the example above. In line with expectations, MPC 3.4 reaches infeasibility at the deceleration peak, thanks to the hard terminal-set constraint, while MPC 3.3 is able to venture back onto the $s(k+N_\text{p}|k)+\beta v(k+N_\text{p}|k)-s_G = 0$ line and continue to converge towards the goal position equilibrium. In Figure 3.9, parameters are tuned way below the appropriate, $\beta = (-v_{\max}/a_{\min} - \frac{\delta t}{2}) * 0.2$ s, 20% of what would be needed for positive-control-invariant terminal set. MPC 3.4 remains infeasible but MPC 3.3 would exploit its built-in softness and since it is not able to return to the left side of the $s(k + N_\text{p}|k) + \beta v(k + N_\text{p}|k) - s_G = 0$ line, it passes over the goal position before fully stopping. The cost evolution for this simulation is shown in Figure 3.10 where MPC 3.3 can no longer reach zero cost due to final error relative to the

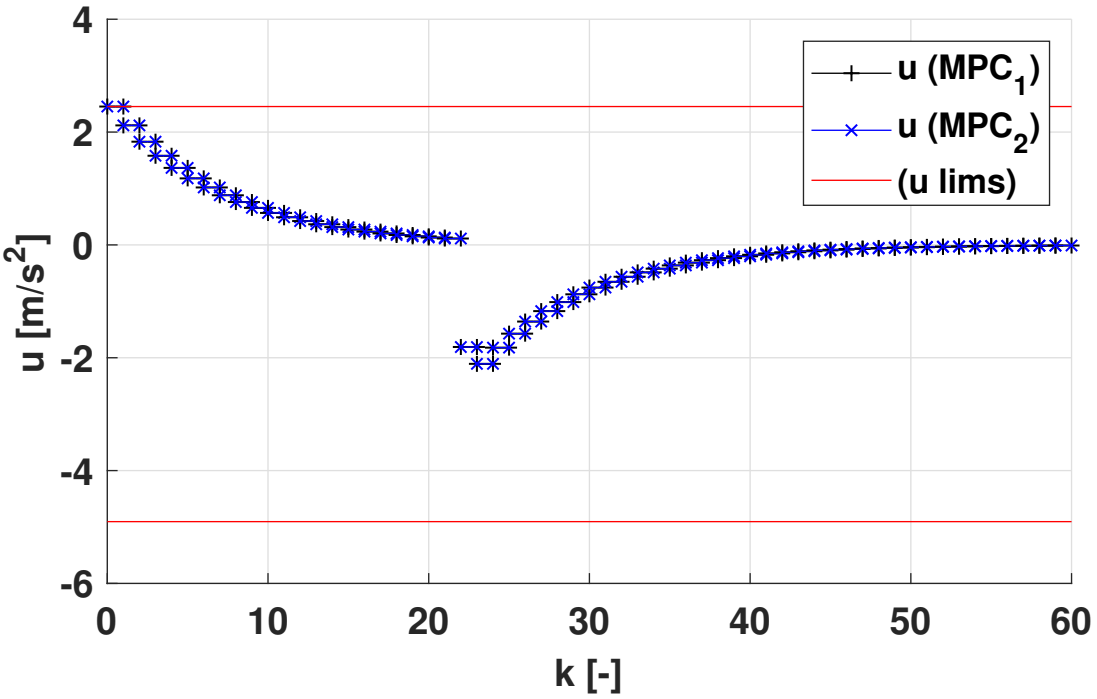
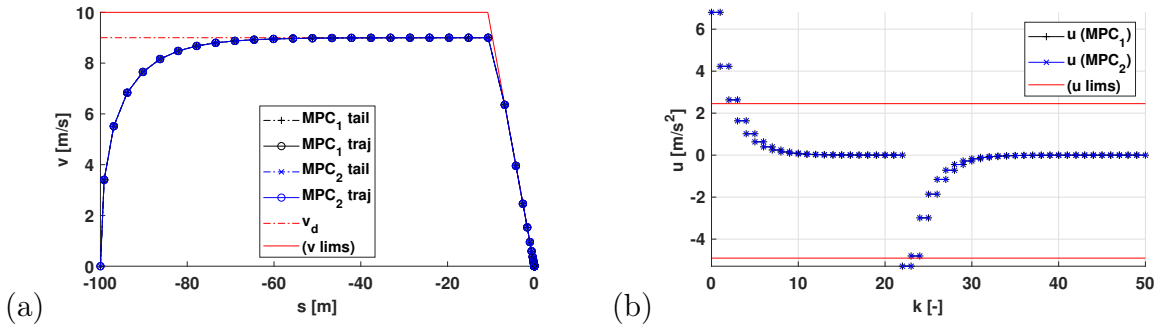


Figure 3.6: Coinciding MPC 3.3 (MPC_1) and MPC 3.4 (MPC_2) control inputs (accelerations).



(a) (b)

Figure 3.7: Comparison of the two MPC controller trajectories: MPC 3.3 (MPC_1) and MPC 3.4 (MPC_2), when β is 60% of what is needed for positive-control invariance: **(a)** v–s and **(b)** u–k.

goal position.

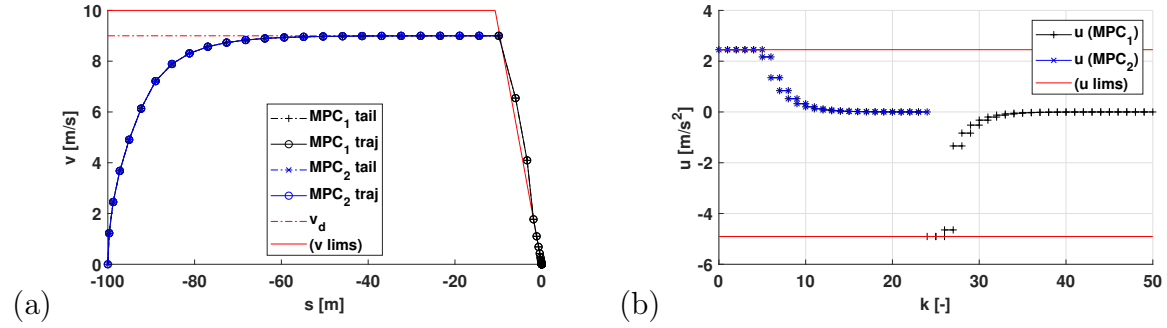


Figure 3.8: Comparison of the two MPC controller trajectories: MPC 3.3 (MPC_1) and MPC 3.4 (MPC_2), when the input-output limit constraints are enabled in both formulations and β is 60% of what is needed for positive-control invariance: **(a)** v–s and **(b)** u–k.

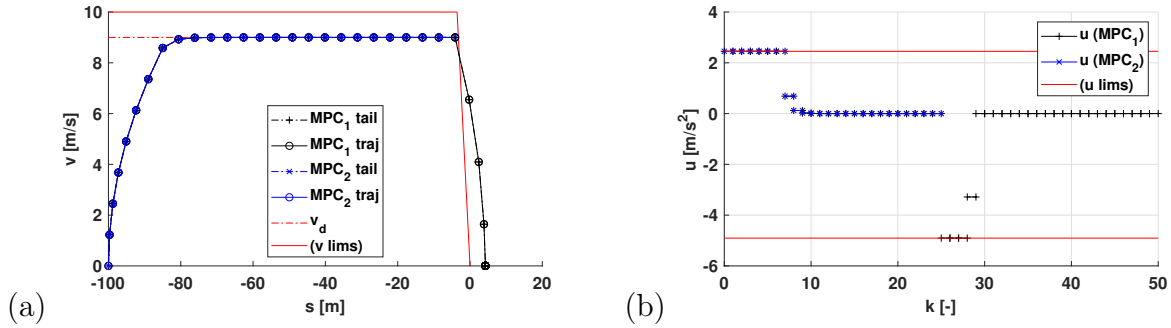


Figure 3.9: Comparison of the two MPC controller trajectories: MPC 3.3 (MPC_1) and MPC 3.4 (MPC_2), when the input-output limit constraints are enabled in both formulations and β is 20% of what is needed for positive-control invariance: **(a)** v–s and **(b)** u–k.

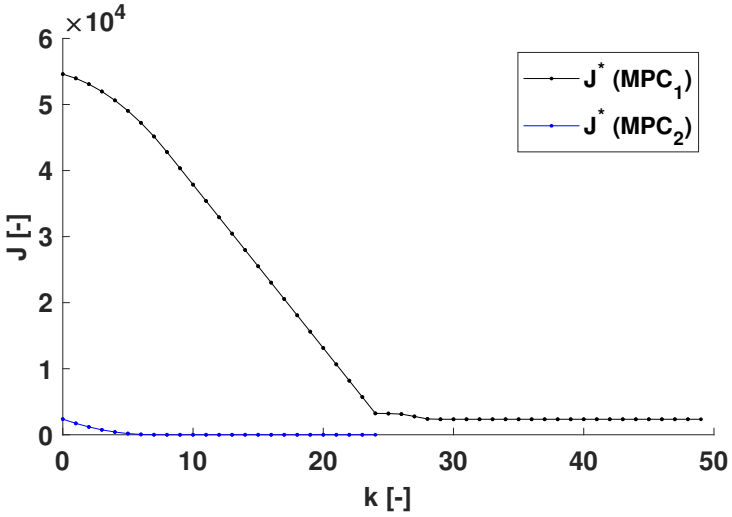


Figure 3.10: Cost comparison of the two MPC controller formulations: MPC 3.3 (MPC_1) and MPC 3.4 (MPC_2), when the input-output limit constraints are enabled in both formulations and β is 20% of what is needed for positive-control invariance.

3.4 Slow-down effect of multiple step horizons

Horizons with $N_\mathrm{p} > 1$ have early reactions to known obstructions (or goal position) which represent, in these cases, the right side of terminal set for hard formulations. Furthermore, the two test formulations are changed to have the speed and acceleration limits enabled.

MPC-3.5, MPC formulation with l_1-norm based terminal-cost

$$
\begin{aligned}
J^*_{\text{MPC-3.5}} = \min \; & \sum_{j=0}^{N_\mathrm{p}-1} \left(q v(k+j|k)^2 + r u(k+j|k)^2 \right) + q_\mathrm{f} v(k+N_\mathrm{p}|k)^2 \\
& + \rho \left| s(k+N_\mathrm{p}|k) + \beta v(k+N_\mathrm{p}|k) - s_G \right| \\
\text{s.t.} \quad & \forall j \in \{0 \dots (N_\mathrm{p}-1)\} : \\
& x\,(k|k) = \; x\,(k) \\
& x\,(k+j+1|k) = \; A\; x\,(k+j|k) + B\; u\,(k+j|k) \\
& x \in \mathcal{X}, \qquad u \in \mathcal{U}
\end{aligned}
$$

MPC-3.6, MPC formulation with 'running-cost' and terminal set

$$
\begin{aligned}
J^*_{\text{MPC-3.6}} = \min \; & \sum_{j=0}^{N_\mathrm{p}-1} \left(q \xi(k+j|k)^2 + r u(k+j|k)^2 \right) + q_\mathrm{f} \xi(k+N_\mathrm{p}|k)^2 \\
\text{s.t.} \quad & \forall j \in \{0 \dots (N_\mathrm{p}-1)\} : \\
& x\,(k|k) = \; x\,(k) \\
& x\,(k+j+1|k) = \; A\; x\,(k+j|k) + B\; u\,(k+j|k) \\
& x \in \mathcal{X}, \qquad u \in \mathcal{U} \\
& s(k+N_\mathrm{p}|k) + \beta v(k+N_\mathrm{p}|k) - s_G \leq 0
\end{aligned}
$$

For simplicity, assume that the controller parameters are chosen to guarantee the recursive feasibility of MPC 3.6; thus, MPC 3.5 and MPC 3.6 formulations yield equivalent mathematical optimisations and optimal trajectories. It is easy to see that the $s(k+N_\mathrm{p}|k) + \beta v(k+N_\mathrm{p}|k) \leq s_G$ constraint from the terminal set acts as an obstruction when active for MPC 3.6 as it causes $v(k+N_\mathrm{p}|k) \to 0$ for operational case $s(k+1) > s(k)$ when $k \to \infty$ making v_d an unreachable terminal setpoint. However, the terminal constraint does not apply for states after the finite horizon, such as $s(k+j|k)$ and $v(k+j|k)$ for $j > N_\mathrm{p}$; thus, the infinite horizon tail can return to v_d (see Figure 3.5). Otherwise the setpoint would be unreachable for the whole infinite prediction horizon, causing the infinite horizon cost integral of MPC 3.6 to be ∞, practically known as infeasible optimisation.

Since the obstruction is present when $s(k+N_\mathrm{p}|k) + \beta v(k+N_\mathrm{p}|k) \leq s_G$ is active, the reaction consist of convergence towards a new coasting speed in plans unique to each

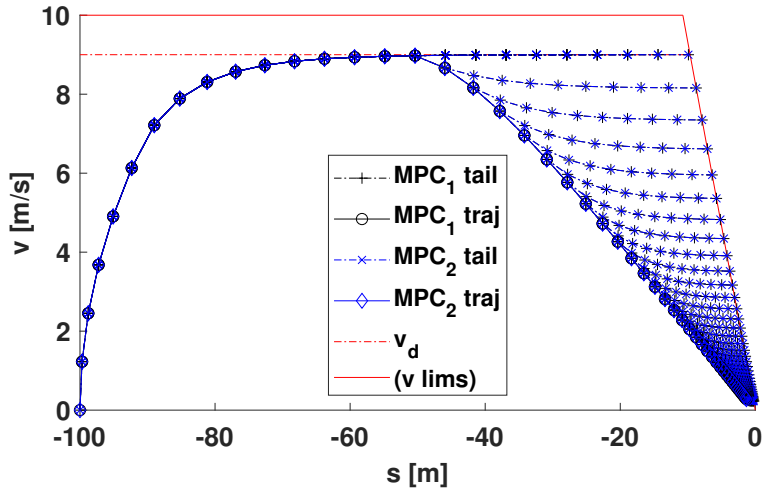


Figure 3.11: MPC 3.5 (MPC_1) and MPC 3.6 (MPC_2) approaching the goal position and gradually slowing down, $N_\text{p} = 10$.

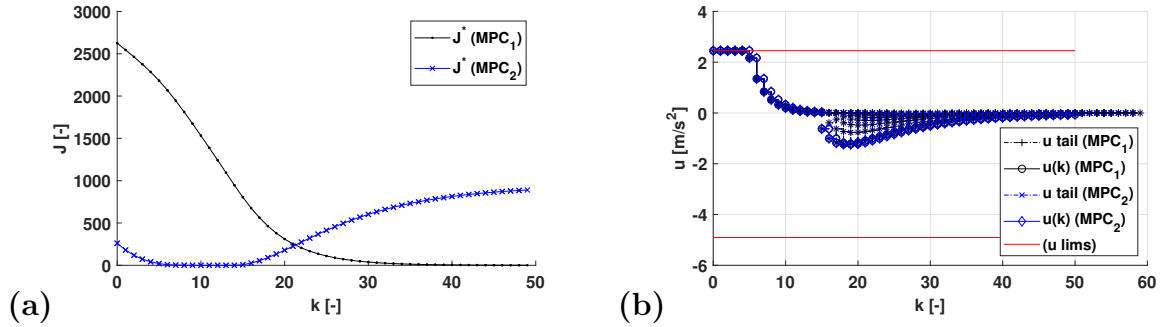


Figure 3.12: Comparison of the two MPC controller formulations: MPC 3.5 (MPC_1) and MPC 3.6 (MPC_2); **(a)** J^* optimal costs and **(b)** u–k.

initial state. This slow-down effect is shown in Figure 3.11; it gives back the same optimal trajectories for both MPC 3.5 and MPC 3.6, respecting the terminal set. In both cases, however, the predicted tail follows only for the first control step. The resulting progress in vehicle position pushes the trajectory closer to the goal position, decreasing the space available for the finite horizon while the length of the time horizon remains the same. This evolution yields a speed decrease between the consecutive prediction tails. Figure 3.12 shows the cost and control input evolutions.

Simulations with higher N_ps (see Figure 3.13) show earlier responses to approaching a goal position, since generally more distance can be covered considering longer horizons. This yields more room for changes and mild control actions to minimise the control cost. The trajectories show, Figure 3.13 **(b)**, convergence towards certain linearly decreasing characteristics after the transient phase for each N_p. To further investigate this, formulate the control policy gain for the first control actions in a manner similar to that for (3.19):

$$\kappa(k) = -\frac{u^*(k|k)}{v(k)}, \tag{3.30}$$

where the calculated equivalent gain is κ. In Figure 3.14, the calculated κ evolutions are

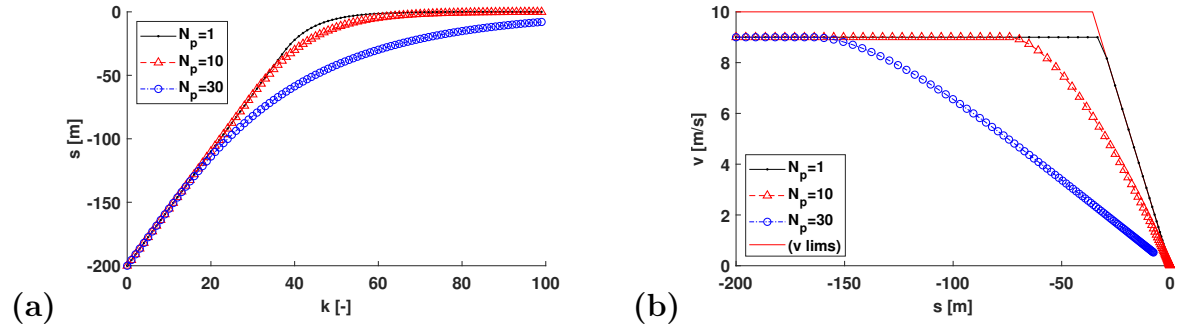


Figure 3.13: **(a)** s–k and **(b)** v–s graphs for goal-approaching simulation with $N_\mathrm{p} = \{1, 10, 30\}$ horizons.

shown for goal approach simulations; their respective trajectories are shown in Figure 3.13. In the case of $N_\mathrm{p} = 1$, the design gain of the terminal set from the β headway time and δt control period time (3.20) is attained because the horizon has no information about the obstruction presented by the terminal constraints until they are reached and activated. κ remains more moderate for higher horizon lengths than short horizons practically limited by the design of the terminal set. Moreover, the control action is applied sooner and tends to converge after a transient phase. A set of horizon lengths are tested where κ values at the end of the simulations are sampled and assumed to be sufficiently converged (close to stop). These κ values are used to calculate control-period-normalised time headways:

$$\frac{\hat{\beta}}{\delta t} = \frac{1}{\kappa \delta t} - \frac{1}{2} \tag{3.31}$$

which are shown in Figure 3.15. Note that, for a low number of horizon steps, the formulation ensures safety by limiting the rate of slow-down through the original β design parameter to safe levels as proved by the converged trends of slow-down trajectories. This limiting effect can be seen in Figure 3.15 as the $\frac{\beta}{\delta t}$ line. For longer horizons, the trajectory profiles converge to the normalised time headway corresponding to $N_\mathrm{p} - \frac{1}{2}$. This shows that with the tuning choices, longer horizons are able to accommodate longer slow-down trajectories to static obstacles or goal positions, inherently respecting control limits, and short horizon formulations are actively benefiting from the safety design choices in the framework.

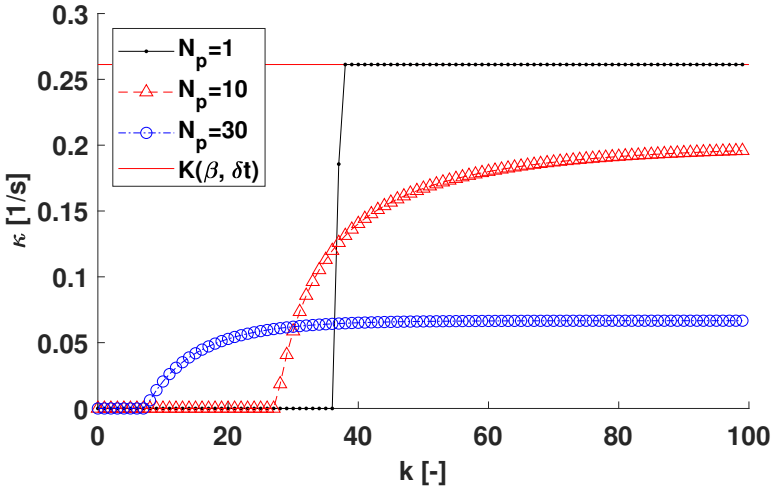


Figure 3.14: The evolution of κ gain for goal-approaching simulation with $N_{\mathrm{p}} = \{1, 10, 30\}$ horizons.

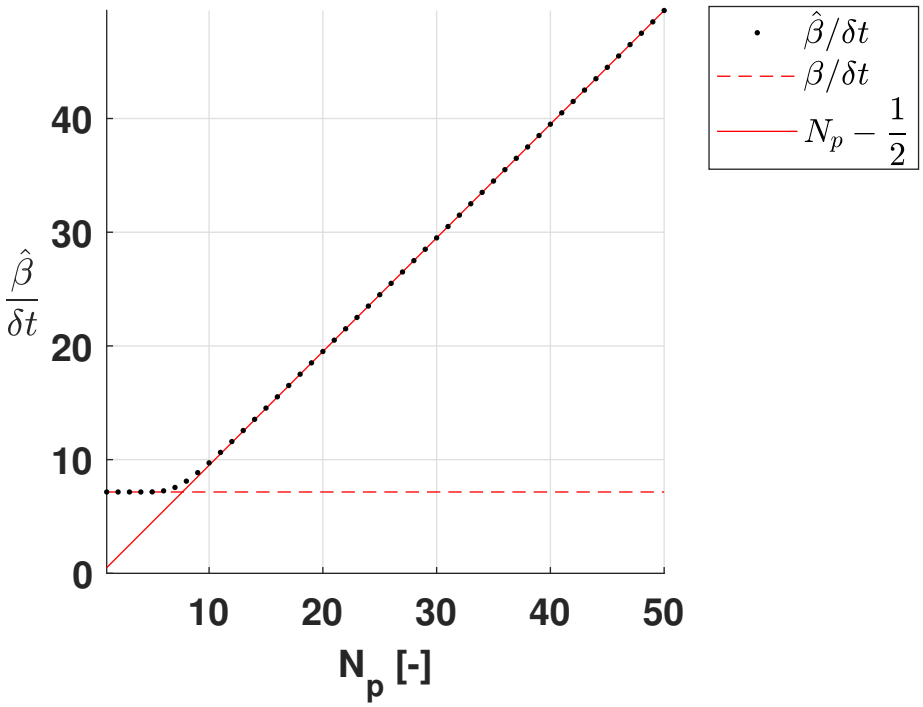


Figure 3.15: The converged and normalised time headway initially follows the design gain of the terminal set for short horizons but switches over to the empirical $N_{\mathrm{p}} - 1/2$ horizon length-dependent line for longer horizons.

3.5 Two-vehicle pre-merging

In the case of car-following, the vehicles have the previously discussed (Section 2.3.4) separation requirement formulated as a hard constraint. When two vehicles merge or pass through junctions, their hard separation constraints are active or relaxed in the MIP formulation to preserve safety (Section 2.5). If a cost were to be assigned to a hard constraint then when it is not violated its penalty cost would be zero, when it is violated its penalty cost would be infinite. It stands to reason that this sudden jump could be made smoother, potentially preparing for future obstacles and conflict zones ahead of time. As shown in Section 3.4, the slow-down effect of a longer horizon result in the potential for an earlier reaction in response to an obstruction. This yields mild, less intensive control actions to achieve the output requirements (e.g. deceleration before an obstacle or in merging creating the appropriate separation for car-following). It would be computationally demanding to simply increase the length of the horizon, thus, another approach will be explored. In this section, an additional cost element is formulated that softly encompasses and adds the vehicle-separation constraint of the merging to the objective function.

For simplicity, let the aggregated cost function for a restricted case be investigated with two vehicles a and b:

$$J = \sigma F_{\text{sep}}(x_a\left(k+N_{\text{p}}|k\right), x_b\left(k+N_{\text{p}}|k\right)) + \sum_{n\in\{a,b\}} w_n F(x_n\left(k+N_{\text{p}}|k\right)) + \sum_{j=0}^{N_{\text{p}}-1} w_n l_n\left(x_n\left(k+j|k\right), u_n\left(k+j|k\right)\right), \tag{3.32}$$

where stage cost is $l_n = q\left(v_n - v_{d\ n}\right)^2 + r\ u_n^2$, the relative weights are w_n, and the terminal cost is $F = q_{\text{f}}\ \left(v - v_d\right)^2$. Moreover, the weight for the soft separation penalty is σ and the soft separation penalty is:

$$F_{\text{sep}} = s_b(k+N_{\text{p}}-1|k) - s_a(k+N_{\text{p}}|k) - \beta v_a(k+N_{\text{p}}|k) - L$$

where the index of the leader vehicle is b and the safe car-following separation gap has yet to be achieved; thus, $s_b(k+N_{\text{p}}-1|k) - s_a(k+N_{\text{p}}|k) - \beta v_a(k+N_{\text{p}}|k) - L \geq 0$ and, as such, $F_{\text{sep}} \geq 0$. Later, this formulation is generalised for the whole scenario with arbitrary vehicle positions. Let the positions be reformulated with the use of the speed decision variables with the help of (3.9):

$$\begin{aligned} F_{\text{sep}} = {} & s_b(k) - s_a(k) - L + v_b(k)\ \frac{\delta t}{2} - v_a(k)\ \frac{\delta t}{2} + v_b(k+1|k)\delta t - v_a(k+1|k)\delta t \\ & + \ldots + v_b(k+N_{\text{p}}-1|k)\frac{\delta t}{2} - v_a(k+N_{\text{p}}-1|k)\delta t - v_a(k+N_{\text{p}}|k)\left(\frac{\delta t}{2}+\beta\right) \end{aligned}$$

Following the arguments in Section 3.2, the constants can be disregarded in the optimisation (e.g. $s_b(k) - s_a(k) - L + v_b(k)\ \frac{\delta t}{2} - v_a(k)\ \frac{\delta t}{2}$). Therefore, what remains are linearly dependent speed terms that amend the originally desired vehicle speeds through weighting. The desired speeds are ordered in vector forms for better representation where the elements are desired speed to be considered in stage costs and the terminal speed at $k,\ k+1, \ldots,\ N_\mathrm{p}-1,\ k+N_\mathrm{p}$.

$$\hat{v}_{d\ a} = \left[v_{d\ a} - \frac{\sigma \delta t}{2 w_a q}, v_{d\ a} - \frac{\sigma \delta t}{2 w_a q}, \ldots, v_{d\ a} - \frac{\sigma \delta t}{2 w_a q}, v_{d\ a} - \frac{\sigma \left(\frac{\delta t}{2} + \beta\right)}{2 w_a q_\mathrm{f}} \right]^T,$$

$$\hat{v}_{d\ b} = \left[v_{d\ b} + \frac{\sigma \delta t}{2 w_b q}, v_{d\ b} + \frac{\sigma \delta t}{2 w_b q}, \ldots, v_{d\ b} + \frac{\sigma \delta t}{4 w_b q}, v_{d\ b} \right]^T$$

It should be noted that the desired speed for first time step (k) is arbitrary since the speed state is a fixed value $v(k|k) = v(k)$, thus, as a constant shift it may be also disregarded in the optimisation. Moreover, the leader vehicle has its desired speed increased for ($N_\mathrm{p} \geq 2$) horizons while the follower vehicle has its speed decreased for all valid horizons ($N_\mathrm{p} \geq 1$) with the terminal desired speed having a change compared to that at the middle of the horizon. This would mean that in the case of $N_\mathrm{p} = 1$, the leader vehicle experiences no change in its cost and makes no effort to cooperate with the follower vehicle (Figure 3.16).

Moreover, this formulation would cause the desired speeds to be changed even after the desired separation is achieved (see Figure 3.16). Formulating it as a true soft penalty, the cost can be deactivated when the intended separation is reached.

MPC-3.7

$$\begin{aligned}
J^*_{\text{MPC-3.7}} = \min\ & \sigma F_\mathrm{sep} + \sum_{n \in \{a,b\}} w_n F(x_n\,(k+N_\mathrm{p}|k)) \\
& + \sum_{j=0}^{N_\mathrm{p}-1} w_n l_n\,(x_n\,(k+j|k)\,, u_n\,(k+j|k)) \\
\text{s.t.}\quad & \forall j \in \{0 \ldots (N_\mathrm{p}-1)\},\ \forall n \in \{a,b\}: \\
& x_n\,(k|k) = x_n\,(k) \\
& x_n\,(k+j+1|k) = A\ x_n\,(k+j|k) + B\ u_n\,(k+j|k) \\
& x_n \in \mathcal{X}_n \\
& u_n \in \mathcal{U}_n \\
& s_a(k+N_\mathrm{p}|k) + \beta v_a(k+N_\mathrm{p}|k) - s_b(k+N_\mathrm{p}-1|k) + L - F_\mathrm{sep} \leq 0 \qquad (3.33a) \\
& F_\mathrm{sep} \geq 0 \qquad (3.33b)
\end{aligned}$$

When the soft separation cost is inserted into the merging scenario, it is beneficial to see that the car-following modes utilise the same hyperplane. Practically, it is a linear

soft penalty version of the later-appearing hard constraints promoting a softer transition between the approaching and car-following modes. When the states reach the hard car-following constraints of the same kind this F_{sep} cost would turn to zero and vanish from the optimisation. σ weight sets the intensity of the earlier response while the timeliness of completion depends on the intensity as well. This means that late information may not lead to significantly different trajectories while too soon knowledge would yield the same fixed rate of trajectory change. In the simplest case of two-vehicle merging, this cost has to be defined for both vehicle orders and, with the already existing binaries, the extra penalty terms may be relaxed just like a hard constraint would.

$$J = \sigma \left(F_{\text{sep } ab} + F_{\text{sep } ba}\right) + \sum_{n\in\{a,b\}} w_n F(x_n\left(k+N_\text{p}|k\right)) + \sum_{j=0}^{N_\text{p}-1} w_n l_n\left(x_n\left(k+j|k\right), u_n\left(k+j|k\right)\right), \tag{3.34}$$

Furthermore, the additional constraints are:

$$s_a(k+N_\text{p}|k) + \beta v_a(k+N_\text{p}|k) - s_b(k+N_\text{p}-1|k) + L - F_{\text{sep } ab} \leq M\ b_1(k+N_\text{p}|k) \tag{3.35a}$$

$$s_b(k+N_\text{p}|k) + \beta v_b(k+N_\text{p}|k) - s_a(k+N_\text{p}-1|k) + L - F_{\text{sep } ba} \leq M\ b_2(k+N_\text{p}|k) \tag{3.35b}$$

$$F_{\text{sep } ab} \geq 0, \tag{3.35c}$$

$$F_{\text{sep } ba} \geq 0, \tag{3.35d}$$

where the binary variable connected to obstacle-approaching mode in merging is $b_1(k+N_\text{p}|k)$ for vehicle a (which mode is active for $b_1(k+N_\text{p}|k) \equiv 0$) and similarly the binary variable connected to obstacle-approaching mode in merging is $b_2(k+N_\text{p}|k)$ for vehicle b (which mode is active for $b_2(k+N_\text{p}|k) \equiv 0$). The obstacle-avoidance constraints and their handover for merging were discussed in Section 2.5; the corner-cutting prevention was discussed in Section 2.3.5.1.

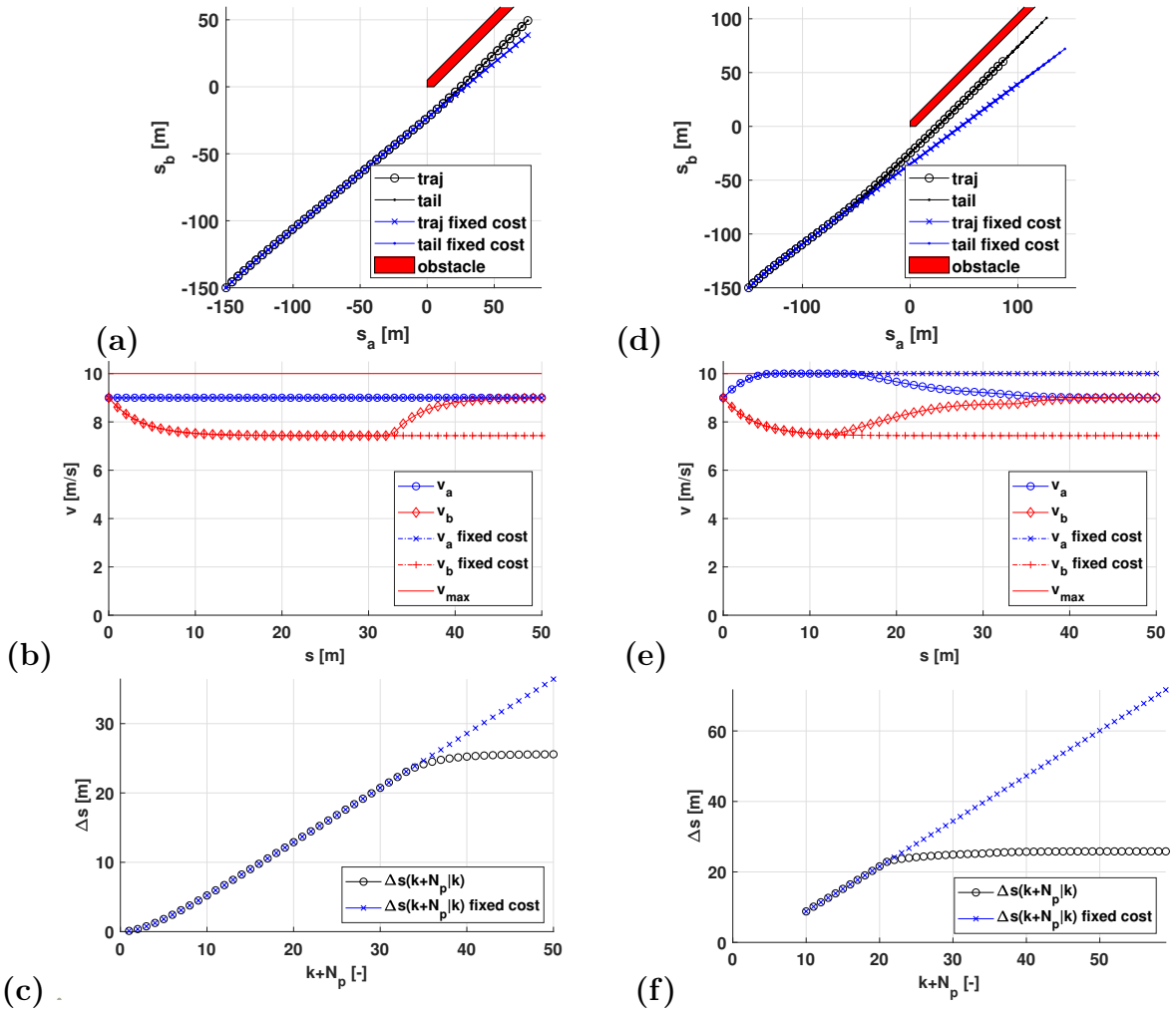


Figure 3.16: **(a)**, **(b)** and **(c)** show simulations for $N_\mathrm{p} = 1$ horizon length where the leader vehicle is not cooperating; **(d)**, **(e)** and **(f)** show simulations for $N_\mathrm{p} = 10$ horizon length where the leader vehicle is cooperating. Two formulations are compared: one is where the cost is in constraint penalty form MPC 3.7; the other simulation is for the original fixed separation cost example (3.32).

MPC-3.8

$$
\begin{aligned}
&J^*_{\text{MPC-3.8}} = \min\ (3.34) \\
&\text{s.t.} \quad \forall j \in \{0 \dots (N_\mathrm{p} - 1)\},\ \forall n \in \{a, b\} : \\
&\qquad x_n\,(k|k) = x_n\,(k) \\
&x_n\,(k + j + 1|k) = A\ x_n\,(k + j|k) + B\ u_n\,(k + j|k) \\
&\qquad\qquad x_n \in \mathcal{X}_n \\
&\qquad\qquad u_n \in \mathcal{U}_n \\
&\qquad\qquad [x_a, x_b] \in \Omega^{\text{ccp,st}}_{ab}, \\
&\qquad\qquad (3.35a)\text{–}(3.35d)
\end{aligned}
$$

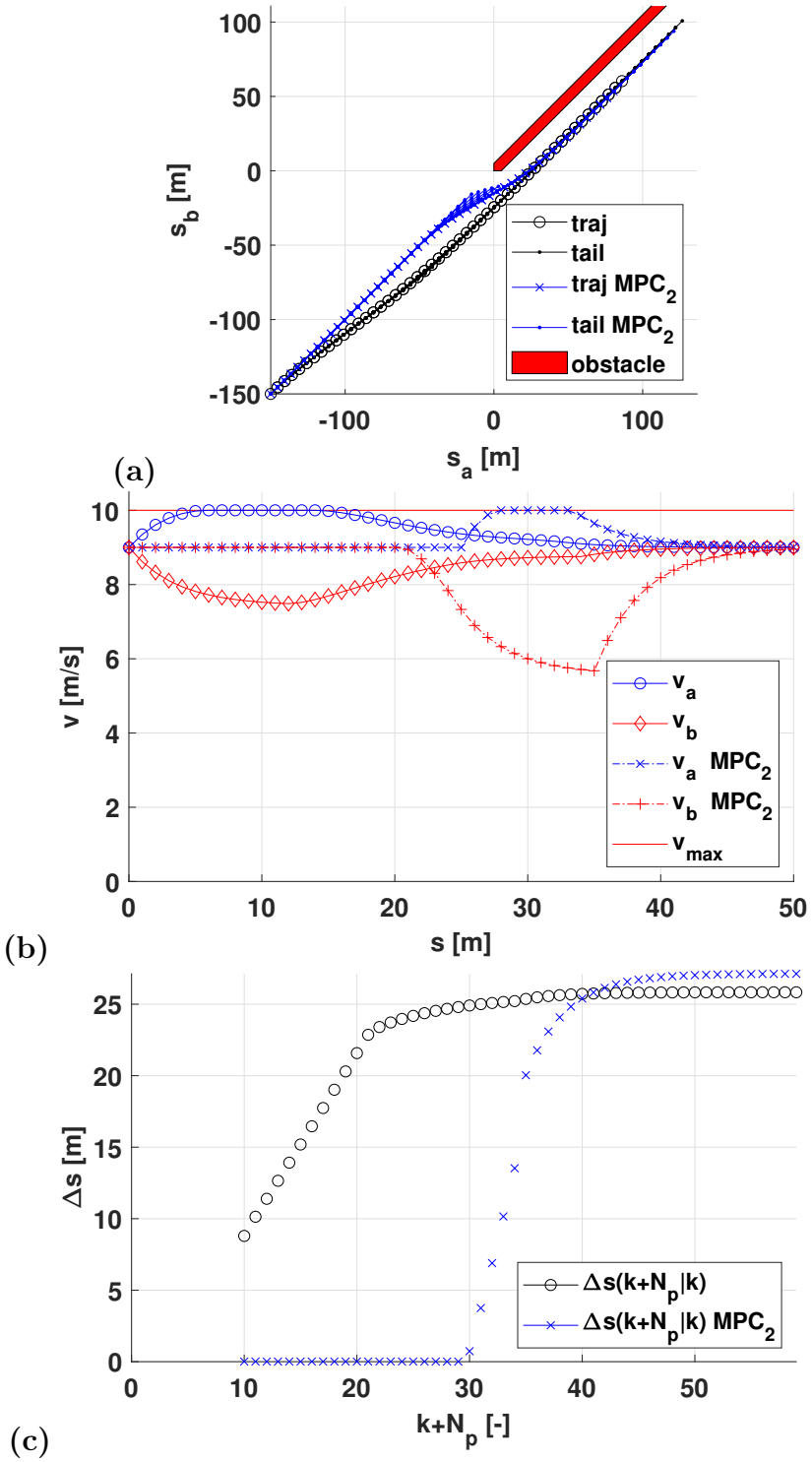


Figure 3.17: The new pre-merging formulation, MPC 3.8, is compared with MPC 3.6 (MPC_2) formulation with added vehicle-merging constraints for $N_\text{p} = 10$ horizon length. The new formulation immediately reacts to the information about the obstacle. In contrary, without the soft pre-merging constraints the controller only senses the obstacle when its horizon comes in contact with the obstacle-avoidance hard constraints.

3.6 Junction speed limits

It is useful in some cases to define a restriction region in the junction with a new maximum speed limit and/or a non-zero minimum speed requirement. Reduced maximum speed limit can enhance safety in busy junctions or on curvilinear paths by limiting the centrifugal forces acting on vehicles and, thus, preventing dangerous roll-over situations. This speed limit can be set with respect to passenger comfort, cargo safety, or lateral accelerations. When the osculatory circle is in a plane parallel to the ground, the lateral acceleration may be approximated with the normal component of total acceleration:

$$a_{lat\ \max} \geq a_{lat} \approx a_n = \frac{v^2}{R},$$

where the speed is v and the radius of the osculatory circle is R (or, in the approximated sense, the turning radius). Thus, the upper speed limit may be calculated for known radii:

$$v_{\max}(R) \approx \sqrt{R\ a_{lat\ \max}}$$

The non-zero minimum speed requirement in junction areas promotes finite occupation time of the shared road section. Even small junction may be susceptible to deadlocks formed when multiple vehicles occupy the junction and blocking the traffic. In [3], no-stop regions were implemented with MIP tools to keep vehicles moving in junctions; the authors added additional speed-ramp-up space where fully stopped vehicles were guaranteed to be able to reach the needed minimum speed required in the region.

With the help of big-M relaxations minimum and maximum speed limits are implemented as follows:

$$s(k+j|k) + t_{\mathrm{h}} v(k+j|k) \leq s_{r2} + M_{sv} b_1(k+j|k) \tag{3.37a}$$
$$v(k+j|k) \leq v_{r\max} + M_v b_2(k+j|k) \tag{3.37b}$$
$$-s(k+j|k) \leq s_{r3} + M_{sv} b_3(k+j|k) \tag{3.37c}$$
$$s(k+j|k) - t_{h\ low} v(k+j|k) \leq s_{r1} + M_{sv} b_1(k+j|k) \tag{3.37d}$$
$$-v(k+j|k) \leq -v_{r\min} + M_v b_2(k+j|k) \tag{3.37e}$$
$$\sum_{i=1}^{3} b_i(k+j|k) \leq 2, \tag{3.37f}$$

where $j = \{1, \ldots, N_{\mathrm{p}}\}$ and the start position for the speed-up ramp is s_{r1}, the projected position for the maximum speed time-headway constraint is s_{r2} and the junction region end position is at s_{r3}. Notice that $M_v < M_{sv}$ is a tightening opportunity in the speed big-M relaxations since the speed range is bounded $v \in [0, v_{\max}]$, yielding $M_v = v_{\max}$, which followed (2.47), as in [8].

For the time-headway parameters the conditions are $\delta \mathrm{t} \leq 2\mathrm{t_h}$, $\mathrm{t_h} \geq \frac{v_{\max}}{-a_{\min}} - \frac{\delta \mathrm{t}}{2}$, $\mathrm{t_{h\,low}} \geq \frac{v_{r\,\min}}{a_{\max}}$. As previously discussed, corner-cutting may occur in inter-samples (see Figure 3.18 **(a)**) which, for long control period times, may disregard the full obstacle or restriction. For corner-cutting prevention, the same hyperplanes are to be defined at the neighbouring time steps while the relaxation binaries are the same as those for the original hyperplanes [63].

$$s\,(k+j-1|k) + t_{\mathrm{h}} v\,(k+j-1|k) \leq s_{r2} + M_{sv} b_1(k+j|k) \tag{3.38a}$$
$$v\,(k+j-1|k) \leq v_{r\,\max} + M_v b_2(k+j|k) \tag{3.38b}$$
$$-s\,(k+j-1|k) \leq s_{r3} + M_{sv} b_3(k+j|k) \tag{3.38c}$$
$$s\,(k+j-1|k) - t_{h\ low} v\,(k+j-1|k) \leq s_{r1} + M_{sv} b_1(k+j|k) \tag{3.38d}$$
$$v\,(k+j-1|k) \leq -v_{r\,\min} + M_v b_2(k+j|k) \tag{3.38e}$$

The overall controller is shown in MPC 3.9.

MPC-3.9

$$J^*_{\text{MPC-3.9}} = \min \sum_{j=0}^{N_{\mathrm{p}}-1} \left(q\xi(k+j|k)^2 + ru(k+j|k)^2 \right) + q_{\mathrm{f}}\xi(k+N_{\mathrm{p}}|k)^2$$

s.t. $\forall j \in \{0 \ldots (N_{\mathrm{p}}-1)\}$:

$$x\,(k|k) = x\,(k)$$
$$x\,(k+j+1|k) = A\,x\,(k+j|k) + B\,u\,(k+j|k)$$
$$(x,u) \in \mathcal{X} \times \mathcal{U}$$

with :

(3.37a)–(3.37f) and

(3.38a)–(3.38e).

The handover artefacts can be observed in Figure 3.18 **(b)**, which respect the speed constraints for $N_{\mathrm{p}} = 1$ (i.e. one last section following the time-headway hyperplane before it switches to the middle section with the constant speed limits. This may be an issue if the speed change in one time period is long enough to penetrate the other restricted state region. To avoid this, a sufficient gap should be designed between the speed limits as:

$$\frac{v_{r\,\max} - v_{r\,\min}}{\delta t} \geq \min(\mathrm{t_h}, \mathrm{t_{h\,low}}).$$

The handover artefacts are less dominant for longer horizons; an example is shown for $N_{\mathrm{p}} = 5$ in Figure 3.18 **(c)**.

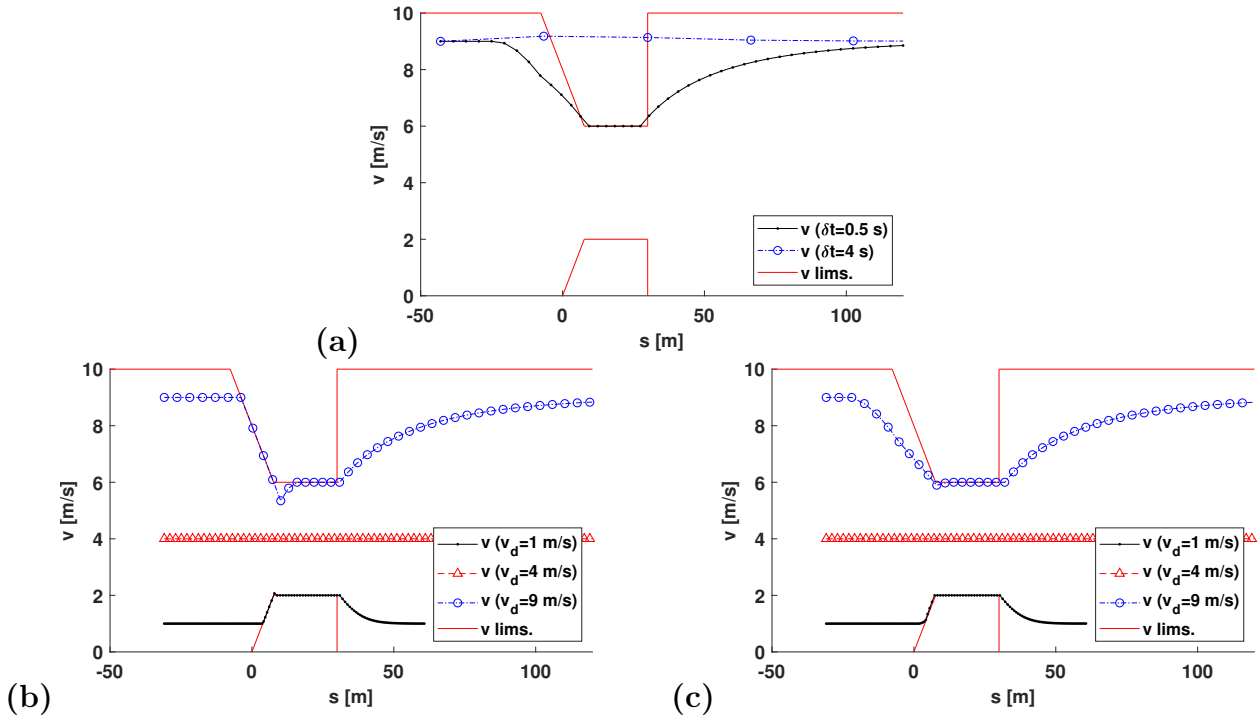


Figure 3.18: **(a)** shows simulation trajectories without corner-cutting prevention for $N_\mathrm{p} = 5$ where the solutions cut into the speed limits, a particularly dangerous cut can happened for long period times (such as $\delta t = 4$ s). In **(b)**, corner-cutting prevention is in place with $N_\mathrm{p} = 1$ horizon length showing the artefact peaks between hyperplane handovers; this effect is mitigated for longer horizons, as shown in **(c)** with $N_\mathrm{p} = 5$.

4

Cross-junction control and simulations

This chapter aims to expand the control framework to higher-scale simulations, increasing the simulation length and the number of vehicles considered in the multi-agent models. The chosen test environment is a single simplistic cross junction, where complexity comes from the junction-passing arbitration between vehicles. After a brief introduction to the numerical considerations for these simulations, the considered types of junction simulations and vehicle interactions will be discussed. Various control policies are introduced, formulated, and analysed in the context of junction examples; further investigation is made into deadlock situations. The junction-passing completion approach is formulated for the so-called 'box junctions' using MIP techniques, which aim to solve deadlock issues that are common in busy intersections.

4.1 Problem statement

The number of agents (vehicles) within the system is N, moving on two junction arms. Let the west-to-east arm be referred to as arm_1 and the south-to-north arm as arm_2 with the number of vehicles on them N_{arm_1} and N_{arm_2}, respectively. On Figure 4.1, the vehicle parameters are shown with the bounding box where the length parameter is L with an added safety gap (vehicle length is $\hat{L}$) and the width parameter is W, resulting in the pictured collision set $\mathcal{C}_{12}$.

The vehicles move on their respective junction arms and must remain safe while passing through the junction. As before, the vehicles are to optimise some measure of their objective and cooperate through centralised optimisation.

Cross junctions are inherently susceptible to gridlock or deadlock situations. In computer science, 'deadlock' situation is a state in which two or more processes have tasks competing for a shared resource that can only be used by one at a time; the processes ultimately block each other, resulting in neither gaining access to the resource or completing their task in finite time without outside help. This state may be resolved through

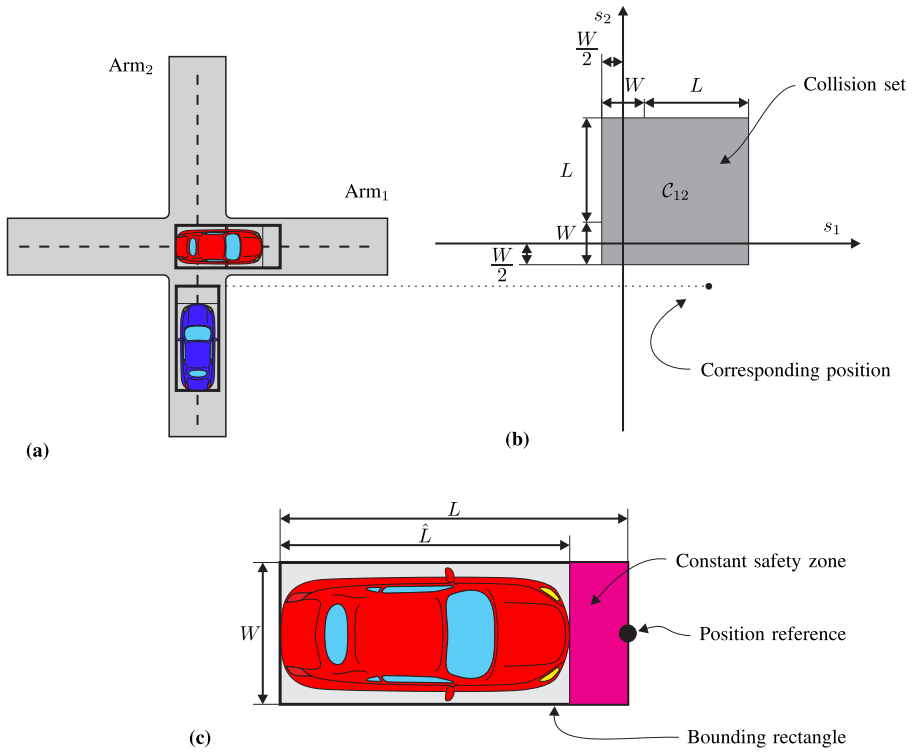


Figure 4.1: **(a)** Schematic of a cross junction with orthogonal arms; **(b)** the constructed collision set for a conflicting vehicle pair (the pictured scenario on **(a)** is indicated as the corresponding state position on **(b)**); **(c)** the rectangular bounding box design used with indicated vehicle parameters, including a constant safety zone in front of the vehicle—the foremost longitudinal position of the bounding box serves as vehicle's reference point.

additional or external action; without it, however, the system is paralysed indefinitely. Using this analogy, the vehicles moving through the junction are the tasks competing for the shared resource. For example, a greedy driving policy makes a vehicle enter the junction even when it is clear that it may not leave it for an unknown amount of time. Of course, in practice, traffic deadlocks and gridlocks are resolved in finite time due to either outside help or the creative thinking of human drivers.

In comparison, Y-junction merging is structurally deadlock-free; a single greedy vehicle on one arm does not indefinitely obstruct traffic on the other arm, as it leaves the junction.

The box junction also has the benefit of being free of deadlocks, as it restricts entry. According to the UK Highway Code [77] for box junctions the indication is as follows: 'Box junctions have criss-cross yellow lines painted on the road (...) You MUST NOT enter the box until your exit road or lane is clear.' [77, Rule 174], with some minor condition on right turning vehicles.

4.2 Numerical considerations

The main simulation options, including the choice of cost function, modifications on dynamics, and the simulation type are explained in the following sections.

4.2.1 Choice of cost function

Chapter 3 discussed the so-called 'running-cost' and terminal l_1-norm quadratic cost-based formulations. The terminal l_1-norm cost has the advantage of soft formulation regarding the goal or obstacle. However, the cost value incorporates the distance-to-go until this target position, which can add a large numeric shift relative to that of the 'running-cost', when a vehicle is far from its destination. Moreover, calculating the l_1 norm has the disadvantage of non-smoothness at the apex point (with an ambiguous gradient that the optimisation relies on). In contrast, the equivalent 'running-cost' has a position-dependent hard constraint limit with no softness regarding output states; thus, attention must be paid to the parameter settings of the controller to retain recursive feasibility. An advantage of the 'running-cost' formulation is that cost is not dependent on the distance-to-go. However, it encodes this information with the use of a hyperplane, a hard constraint enforcing the safe terminal set. During pre-solve phase, in case of long distance-to-go, the solver may deem the hyperplane of the hard constraint outside of the reachable set—never to be activated—during the solve phase and, as such, remove it early on as part of the pre-solve. Altogether, the 'running-cost' with hard constraints was chosen because it was found to provide more accurate results; the increased numerical accuracy stems from the numerically better posed optimisation. Thus, the cost function has the following aggregated multi-agent form:

$$J = \sum_{n\in\mathcal{N}} w_n \left(\sum_{j=0}^{N_\mathrm{p}-1} \left[q\xi_n(k+j|k)^2 + ru_n(k+j|k)^2 \right] + q_\mathrm{f}\xi_n(k+N_\mathrm{p}|k)^2 \right), \tag{4.1}$$

where the set of vehicles is $\mathcal{N}$ and the speed reference error of agent n is $\xi_n = v_n - v_{\mathrm{ref}n}$; the finite horizon length (number of steps) is N_p; the LQR weights are q and r, the terminal weight is q_f; the relative weight of vehicle n is w_n, expressing the relative cost contribution of each vehicle to the aggregated objective function.

4.2.2 Discontinuous dynamics

This section highlights the fact that the simplistic double integrator of LTI dynamics is on the limit of stability and could have major numerical impacts in simulation loops. This nature of the simulation can gradually drive states into constraint conflicts and control infeasibility. The proposed solution is a simple discontinuity, a dead zone added to the simulation of dynamics without changing the related LTI dynamics within the MPC. This approach was chosen over the alternatives of changing the terminating threshold of the solver or softening the hard constraints.

Solver numerics, noise and model choice lead to infeasibility in certain cases of multi-agent car-following in which vehicles are closely packed and nearing a complete stop (very low-speed constraint knock-on effect). Generally, the shape of the terminal set allows

vehicles to asymptotically converge towards zero speed without ever reaching it in finite time. This may also happen in a long queue of vehicles waiting for right of way, be it at a junction or behind a road blockage in a deadlock situation. After a certain amount of time has elapsed, the optimisation may encounter infeasibility, which seemingly contradicts the theory that has been formulated so far. However, this issue can be traced back to the type of idealised kinematics rule described by the double integrator. Matrix A in (2.8) has $\lambda = 1$ double eigenvalues placed; with the nature of integrators come the accumulation of negligible numerical noise and imperfections over a long period.

In the case of asymptotically converging speed, the control input (acceleration) is never truly zero while the inherent numerical noise is present. This noise is relatively small and originates from complex solving procedures and finite precision arithmetic errors. For vehicles in motion, with higher-than-negligible speed, the controller has enough room to accommodate the small imperfections—this numerical noise. In contrast, tightly packed vehicle queues have few feasible state choices available because the hard constraints, defined for each vehicle, move closer together, leaving less and less space to accommodate the integrated noise in each control step. This effect eventually causes the hard constraints to be violated and to exceed the solver infeasibility tolerance threshold.

In order to solve this infeasibility issue, the origins of the numerical noise must be well understood. One source of the noise is the termination of the solver algorithm itself. In convex problems, the solver provides the optimal solution and the certificate of global optimality (zero duality gap). In practice, however, this solution is arbitrary within the vicinity of the absolute optimum, where the terminating conditions of the algorithms are satisfied. Note the similar case of termination for the integer decisions in the MIP problem, where the algorithm may terminate early at the point at which it satisfies all threshold conditions. This can be advantageous in the branch-and-bound algorithm, trading MIP optimality gap (slight sub-optimality) for higher computation speed. In other words, this may result in two mathematically and numerically ideal problem yielding different non-unique results, where the solver has the full discretion to select which answer it returns despite any of them being valid.

Tightening the default threshold criteria of the solver—which, to a point, results in more accurate solutions—is a naïve approach, as the improvement is disproportional. The approach increases the number of iterations necessary to reach the termination conditions and the solution. Of course, this has a negative effect on the computation time of the optimisation. However, changing threshold settings cannot remove the inherent digital noise from the floating-point representation; in practice, it defines a natural lower limit of accuracy and a minimum value of threshold parameters. As expected, higher accuracy increased the queuing time of vehicles before the optimisation inside the controller reached the infeasibility message—verifying that the integrated noise is smaller but ultimately not a solution to the issue.

One typical solution in the literature for implementing control in practice is changing the output hard constraints into soft constraints. Reaching an infeasible state in a controller operating with a real plant is undeniably a grave issue. Well-formulated soft constraints can prevent this state altogether and offer a chance for recovery. The current control framework paired with robust control techniques may achieve earlier vehicle stops through constraint tightening, leaving excess space for noise accumulation but without removing the inherent integration issue of the dynamics (in the simulation loop).

In practice, consider real world vehicle behaviours around obstacles, a full stop phase happens in case of long waits. Since vehicles are not designed or required to operate with continuous motion at very low speeds, they are not equipped to efficiently exert the precise amount of traction force necessary to balance energy dissipation. Furthermore, mechanical systems have increased energy dissipation and discontinuities near stopping. Dry friction from tribology (Coulomb friction) indicates that the coefficient of friction between non-moving surfaces (static friction) is higher than that between moving surfaces (kinetic friction). Moreover, undesirable stick-slip phenomenon in these regions can cause increased wear and tear on vehicles (e.g. gears in the drivetrain, clutch and brakes). Despite it being perfectly desirable to maintain very low speed in the cost function, it has no real benefit. Drivers are either stopped by natural energy dissipation or increased dissipation from engaging their brakes close to the end of available space. Moreover, the stopping position is usually a free choice within a sensible region abiding by safety considerations and driver comfort (e.g. ± 20 cm around a sweet spot).

A discontinuous dynamics formulation to be built into the simulation (proposed below) issues a full stop below a certain threshold mimicking real-world stopping phenomena. The MPC controller, however, remains unchanged with the original LTI dynamics. The arising model mismatch is simply accommodated by the inherent robustness of the proposed safety constraint design from *Theorem 1* and Section 2.5.1.

The proposed vehicle dynamics are changed in the simulation to use candidate states as follows:

$$\begin{bmatrix} s_{\text{cand}}(k+1) \\ v_{\text{cand}}(k+1) \end{bmatrix} = A \begin{bmatrix} s(k) \\ v(k) \end{bmatrix} + Bu(k),$$

$$\begin{bmatrix} s(k+1) \\ v(k+1) \end{bmatrix} = \begin{cases} \begin{bmatrix} s_{\text{cand}}(k+1) \\ v_{\text{cand}}(k+1) \end{bmatrix}, & v_{\text{cand}}(k+1) \geq 10^{-4}\ \text{ms}^{-1} \\ \begin{bmatrix} s_{\text{cand}}(k+1) \\ 0 \end{bmatrix}, & v_{\text{cand}}(k+1) \leq 10^{-4}\ \text{ms}^{-1} \ \text{ and } \ v(k) > 0 \\ \begin{bmatrix} s(k) \\ 0 \end{bmatrix}, & v_{\text{cand}}(k+1) \leq 10^{-4}\ \text{ms}^{-1} \ \text{ and } \ v(k) = 0. \end{cases} \quad (4.2)$$

The threshold setting of $v \geq 10^{-4}$ ms^{-1} by assuming a headway time of 2 s corresponds to the earliest stop of 0.2 mm away from the obstacle point. This, from engineering perspective, is negligible change; provides appropriate margins for a numerically stable, (static) long-queue simulation.

4.3 Simulation types

The simulated scenarios are within a spatially restricted region of the traffic network defined by spatial and temporal boundaries as well as initial conditions that require assumptions about the outside (not simulated) traffic states. This work does not aim to create a perfect replica of real-world traffic; rather, it aims to use the simulations as a means to challenge the proposed control through the choice of boundary assumptions. Three simulation types are introduced in this section to conduct analyses on them. The cases are able to develop continuous traffic, settling to steady-state or periodic traffic flows; moreover, generating vehicle inflows from sampled distributions allow long-term simulations to be run and later obtain broader statistical measures.

This chapter discusses a single road crossing and junction area, providing a mixture of car-following and crossing-order decisions. An appealing type of simulation involves fixing the number of vehicles to yield constant average traffic densities [75]. There is a given number of vehicles travelling on each junction arm; when a vehicle leaves its simulated road at the outlet, it reappears at the inlet. The advantage of this simulation is the constant complexity of the control, as the fixed number of vehicles yields a fixed number of states and decision variables. In traffic simulations, reintroduction of vehicles mimics the self-similar junction scenarios that arise in urban traffic networks. An urban district with largely similar junctions (e.g. a grid) can be simulated via a single cell that is expected to develop realistic inlet and outlet traffic measures. In Section 4.3.2, an 8-loop-shaped simulation type is shown with similarly fixed average traffic densities; this simulation type is later used to analyse average traffic measures, traffic flows and densities with fundamental diagrams [75] as well as further investigate deadlocks stemming from high traffic density and control-policy behaviour.

Finally, the junction arm inlet flows are randomly and independently generated in the third simulation type following a general truncated exponential distribution from Appendix B. The inlets in this case produce random vehicle arrivals with no memory effect of how vehicles leave the simulated cell. This results in the negative effect of varying number of simulated vehicles and the knock-on effect on complexity yet provides random and still intensive interactions for junction arbitration to challenge a control policy.

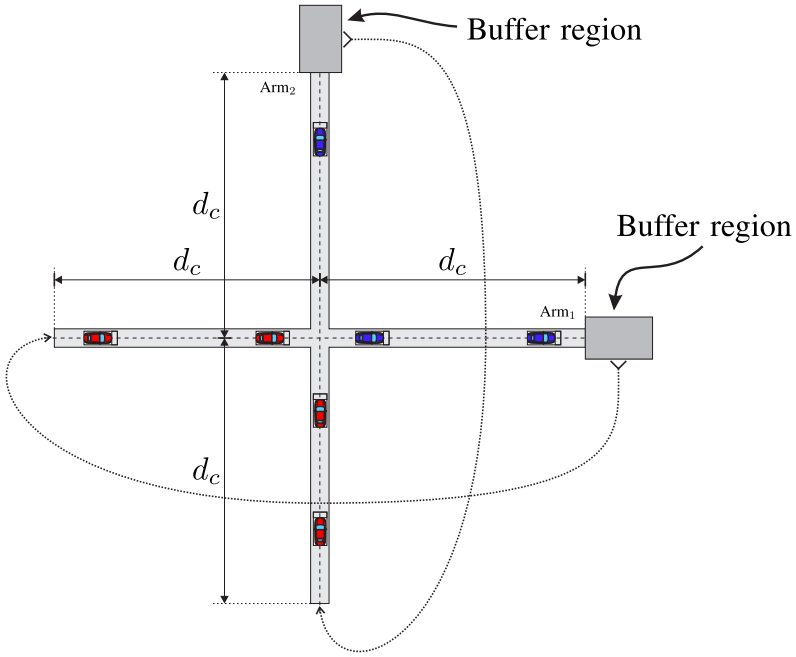


Figure 4.2: Two-arm cross-junction simulation type with road/lane-wise vehicle introduction from the buffer region. This simulation type keeps the number of vehicles, and in turn, the traffic density in the simulation constant. The junction arms each have equal d_c-characteristic arm length.

4.3.1 Fixed number of vehicles—O-loops

First, consider the junction area where a vehicle reaching the end of its road would enter a buffer region. From this buffer region, at each simulation time step, the vehicle is 'reintroduced' as a new vehicle at the beginning of the same road with the same or changed vehicle properties. Such a mechanism is shown in Figure 4.2, where the characteristic arm length is d_c. The buffer area is used to hold vehicles while reintroducing them is unsafe (i.e. the car-following constraint would yield a trivial conflict when the previously reintroduced car has not physically cleared the area yet). Moreover, vehicles have the opportunity to change their states or parameters while reappearing (e.g. changing desired speed).

4.3.2 Fixed number of vehicles—8-loops

When the outlet of arm_1 is connected to the inlet of arm_2 (and vice-versa), an '8-loop' is formed (see Figure 4.3, which shows symmetric loops with the characteristic arm length d_c). In this case, the last vehicle on arm_1 is simply following the first vehicle on arm_2, and vice-versa, removing the need for a buffer region and allowing for seamless vehicle flow. This simulation type can be used to analyse emerging tendencies of long-term simulations in a closed system, for example, steady-state modes, average measures, and stationary solutions.

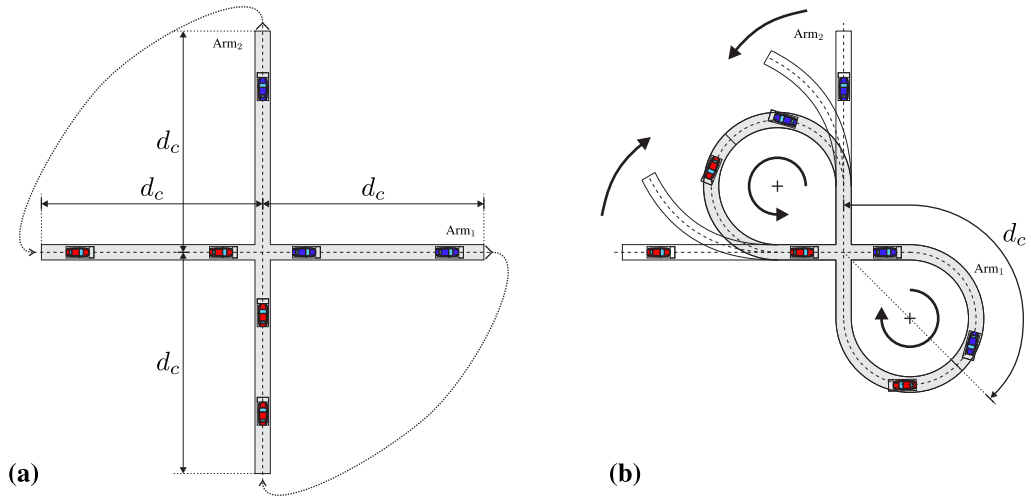


Figure 4.3: 8-loop simulation, in which the vehicles are constantly in transit from the end of one arm to the beginning of the other; **(a)** a simulation schematic and **(b)** the equivalent 8-loop road structure with characteristic arm length d_c.

4.3.3 Junction inlets: Variable number of vehicles

In traffic simulations, vehicle flows may need to be generated to model unrelated and generally asymmetric traffic flows. Figure 4.4 shows how the inlets and outlets are situated on the cross-junction lanes with the characteristic arm length d_c.

According to empirical observations, in uncongested traffic conditions, the number of vehicles passing through a measuring position in each time window follows the Poisson distribution [55]. This distribution is closely related to the negative exponential distribution, which describes vehicle arrival times and defined similarly with one parameter (i.e. the expected value (or mean)). Commonly used distributions to model vehicle flows are Pearson type III, Gamma, Erlang, and negative exponential distributions, each can be derived from the previous one and simpler. In this work, a simple and practical distribution is chosen; a double truncated version of the exponential distribution is used to impose lower and upper limits on vehicle inter-arrival times; in Appendix B the derivation steps, sampling, and an example is shown to generate the vehicle flow. The upper limit removes long inter-vehicle arrival times, keeping the flow vivid. The lower limit is necessarily higher than the t_h time-headway parameter (where t_h, for simplicity, is chosen homogeneously), so that fewer vehicle appear with safety violations. A safety violation can occur at this type of inlet when, for example, a new vehicle would be injected to the traffic simulation, however, this would cause conflicting safety constraint with another vehicle already in the simulation, for example, in case of a saturated queue physically unable to accommodate a new vehicle. If a vehicle can safely appear at the discrete simulation time step according to its designated appearance time, its initial speed will be its desired speed by default, otherwise modified to what the gap headway allows. In contrast, when the injected vehicle would be in violation of constraints, even with stationary state, it is discarded to avoid a collision and control infeasibility. The lower truncation helps

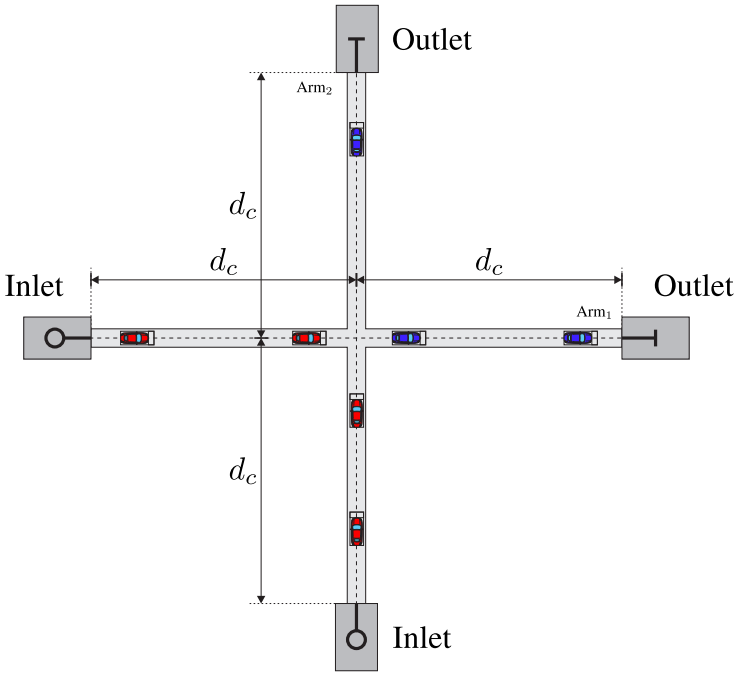


Figure 4.4: This simulation type has vehicle arrivals drawn from a given distribution to introduce vehicles at the inlet port and remove vehicles at the outlet port. The number of vehicles and, in turn, traffic density changes throughout the simulation. The junction arms have equal d_c characteristic length.

to minimise the number of discarded vehicles in normal (uncongested or non-deadlock) operation, which, in turn, better maintains the designed inflow distribution. Finally, the distribution becomes defined by the third parameter: mean flow. More details about the flow generation approach applied here are shown in Appendix B.

4.4 Vehicle interactions

When there is more than one vehicle present on a road network, vehicles may interact with each other to share resources (e.g. roads, lanes, intersection areas). This applies to every scenario in this work. When multiple vehicles are on a single lane, for example, they engage in car-following interactions. When they are on adjacent intersecting roads, they engage in junction arbitration interaction. Figure 4.5 demonstrates, these types of interactions in a simple cross-junction scenario. The continuously reconnected road ends, introduced in Section 4.3.2, are discussed later, because they require more considerations on interactions. On each arm of the junction, vehicles engage in Car-Following Interactions (CFI) with each other. The number of these interactions is N_{CFI}:

$$N_{\mathrm{CFI}} = \max\left(0,\ N_{\mathrm{arm}_1} - 1\right) + \max\left(0,\ N_{\mathrm{arm}_2} - 1\right),$$

where the number of vehicles on junction arm_1 and arm_2 are N_{arm_1} and N_{arm_2}, respectively. For a single road section with $N > 1$ vehicles, the number of car-following interactions is

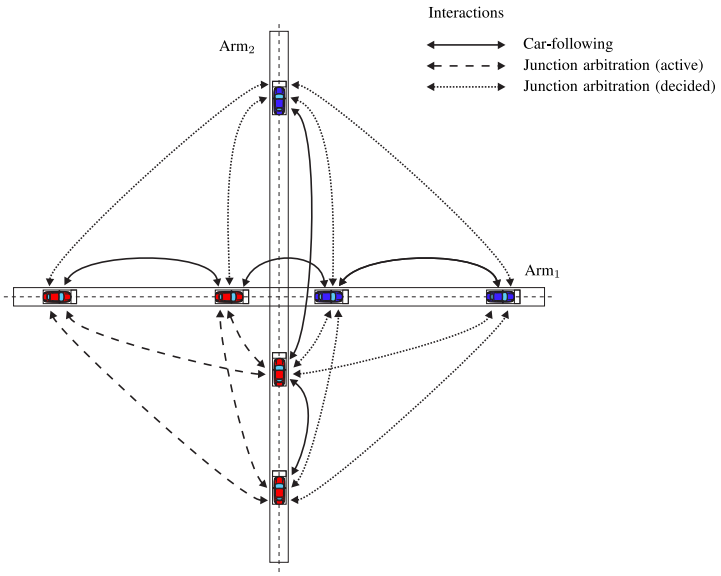


Figure 4.5: Vehicle interactions in a simple two-arm crossing scenario.

$N-1$; $N \equiv 1$ imposes zero car-following pairs.

Car-following constraints were introduced in Section 2.3.4; recall that:

$$s_p\left(k+j|k\right) + t_\mathrm{h} v_p\left(k+j|k\right) \leq s_q\left(k+j+1|k\right) - L_q, \tag{4.3}$$

where the leader vehicle index is q with safety padded length L_q and the follower vehicle index is p with the time headway t_h as an arbitrary choice of the follower with the lower bound depending on the follower vehicle capabilities; $j = \{0, 1, \ldots N_\mathrm{p}\}$. These constraints must be added $\forall\,(p,q) \in \mathcal{N}_\mathrm{CFI}$ vehicle pairs, where the set of all car-following interactions is $\mathcal{N}_\mathrm{CFI}$ composed of the ordered follower–leader subsets (p,q), defined over vehicle indices.

Furthermore, similarly to that in the merging junction, the arbitration-related Junction Crossing Interaction (JCI) in cross junctions can be readily determined, as shown in Figure 4.1. Following this logic, all pairs of vehicles $\{p,q\}$ with JCI between them are collected as subsets of the $\hat{\mathcal{N}}_\mathrm{JCI}$ set. In a simplistic approach, an interaction must be generated for each vehicle on one arm with each vehicle on the other arm (since $\mathcal{C}_{pq} \neq \emptyset$). By repeating this process, the overall number of pairwise junction collision sets is:

$$\hat{N}_\mathrm{JCI} = N_{\mathrm{arm}_1}\ N_{\mathrm{arm}_2}.$$

As shown in Figure 4.5, the junction arbitration corresponding to the interaction of each junction collision set or decision-making process may be either active (waiting for decision) or decided (arbitrated). This property comes from the evolution of states (e.g. when a vehicle has cleared the junction area, the ordering decision can no longer be changed, since the vehicles are not allowed to move backwards). From this consideration, $\mathcal{N}_\mathrm{JCI} \subseteq \hat{\mathcal{N}}_\mathrm{JCI}$ and $N_\mathrm{JCI} \leq \hat{N}_\mathrm{JCI}$, where the set of active junction arbitration interactions is $\mathcal{N}_\mathrm{JCI}$ and the number of vehicle pairs is N_JCI, where neither p nor q has cleared the junction yet. Note

that, an even tighter condition may be obtained if the vehicle pair has fixed arbitration/a decision yielding fewer decisions to be made and, thus, moderate complexity. These fixed decisions can be obtained through policies (e.g. FCFS policy), junction control nodes (Vehicle-to-Infrastructure (V2I) communication), inter-vehicle agreements in decentralised cases (e.g. via Vehicle-to-Vehicle (V2V) communication) or safety considerations of feasible alternatives (e.g. one vehicle has claimed the right of way by its choice of speed). Car-following is also a type of fixed arbitration, because there are no opportunities to change the order in single-lane traffic (with no overtaking allowed); thus, no ordering decisions need to be made because no alternative decisions exist.

Each of the active collision sets (i.e. within $\mathcal{N}_{\mathrm{JCI}}$) shown in Figure 4.5 for cross-junctions requires eight separating hyperplanes and a binary inequality to keep the states of vehicle pairs safe. Thus, the constraints using the simple time-headway parameter and big-M relaxation yields:

$$s_p\left(k+j|k\right)+t_{\mathrm{h}}v_p\left(k+j|k\right)\leq-\frac{W}{2}+Mb_{pq,1}(k+j|k) \tag{4.4a}$$

$$-s_p\left(k+j|k\right)\leq-\frac{W}{2}-L+Mb_{pq,2}(k+j|k) \tag{4.4b}$$

$$s_q\left(k+j|k\right)+t_{\mathrm{h}}v_q\left(k+j|k\right)\leq-\frac{W}{2}+Mb_{pq,3}(k+j|k) \tag{4.4c}$$

$$-s_q\left(k+j|k\right)\leq-\frac{W}{2}-L+Mb_{pq,4}(k+j|k) \tag{4.4d}$$

$$\sum_{i=1}^{4}b_{pq,i}(k+j|k)\leq 3, \tag{4.4e}$$

$$s_p\left(k+j+1|k\right)+t_{\mathrm{h}}v_p\left(k+j+1|k\right)\leq-\frac{W}{2}+Mb_{pq,1}(k+j|k) \tag{4.4f}$$

$$-s_p\left(k+j+1|k\right)\leq-\frac{W}{2}-L+Mb_{pq,2}(k+j|k) \tag{4.4g}$$

$$s_q\left(k+j+1|k\right)+t_{\mathrm{h}}v_q\left(k+j+1|k\right)\leq-\frac{W}{2}+Mb_{pq,3}(k+j|k) \tag{4.4h}$$

$$-s_q\left(k+j+1|k\right)\leq-\frac{W}{2}-L+Mb_{pq,4}(k+j|k), \tag{4.4i}$$

where the first set of constraints (4.4a)–(4.4d) are the binary-relaxed cross-junction hyperplanes, the middle constraint (4.4e) is the separating hyperplane condition and the last four constraints (4.4f)–(4.4i) are the corner-cutting prevention (or inter-sample avoidance) spatio-temporal shifted constraints, where $j=\{0,1,\ldots(N_{\mathrm{p}}-1)\}$.

In summary, the number of interaction constraints is as follows:

$$N_{\mathrm{Icon}}=9N_{\mathrm{JCI}}N_{\mathrm{p}}+N_{\mathrm{CFI}}N_{\mathrm{p}}. \tag{4.5}$$

4.4.1 Simulated region, depth of interaction resolution, and horizon length

The simulation region is the area in the vicinity of the cross junction. In the simulation types discussed in Section 4.3.1 and Section 4.3.3, the vehicles appear in the inlet with restricted knowledge of the traffic state in the junction. Due to the hard constraints, however, the optimisation only allows for safe initial vehicle states when a new vehicle is plugged into the vehicle-control framework. A real-world case would assume that the prediction horizon and perception (e.g. visual information of drivers) are smoothly receding into the traffic area (to some degree, information travels upstream). Thus, in this form, initial vehicle states, spacings (gaps), and other traffic parameters are not optimal at the inlet. Similarly, the closer the vehicle is to the end of its road section and the longer the prediction horizon is, the more that horizon will venture outside of the simulated road section. Since the outside traffic states are un-modelled, they are unknown in terms of vehicles travelling there. This causes the prediction horizon of vehicles, which are following the optimal trajectory, to converge towards their desired speed in these outside regions; this imposes changed behaviours near the end of the road.

In simulation types involving directly connected roads and lanes, inlets and outlets form continuous loops allowing vehicles to seamlessly traverse from one road section to another without defining a buffer policy (e.g. for 8-loop in Section 4.3.2). At the connecting points, however, additional car-following interactions must be added in order for the last vehicle (closest to the outlet) on one road section to respect the first vehicle (closest to where the roads connect) on the other road section. This increases the number of car-following constraints for the 8-loop from Section 4.3.2 with two more car-following constraints:

$$N_{\mathrm{CFI}} = N_{\mathrm{arm}_1} + N_{\mathrm{arm}_2}.$$

The 8-loop example considers a single junction obstacle in the collision set as the original examples. It is recognised that, when the horizon is relatively long compared to the length of the loops it may loop back to the intersection, which in turn would make the collision sets have periodical copies of the same junction obstacles in view. The depth of resolving these interactions is kept only to consider the single closest obstacle in order to limit problem complexity. Furthermore, the simulation types from Section 4.3.1, Section 4.3.2, and Section 4.3.3 share the same consideration about the single closest obstacle resolution, where the differences to the 8-loop example is being the extra car-following constraints and visualisation.

4.5 Policies and examples

This section proposes and discusses policies of baseline control, FCFS control, and pre-avoidance heuristics with numerical examples.

4.5.1 Baseline control policy and examples

The baseline MIQP MPC has the form of:

MPC-4.1

$$
\begin{aligned}
&J^{*}_{\text{MPC-4.1}} = \min\ (4.1)\\
&\text{s.t.}\quad \forall j \in \{0 \dots (N_{\text{p}} - 1)\},\ \forall n \in \mathcal{N}:\\
&\qquad\qquad x_n\,(k|k) = x_n\,(k)\\
&\qquad x_n\,(k+j+1|k) = A\ x_n\,(k+j|k) + B\ u_n\,(k+j|k)\\
&\qquad\qquad (x_n, u_n) \in \mathcal{X}_n \times \mathcal{U}_n\\
&\text{and } \forall (p,q) \in \mathcal{N}_{\text{CFI}}:\\
&\qquad (4.3)\\
&\text{and } \forall \{p,q\} \in \hat{\mathcal{N}}_{\text{JCI}}:\\
&\qquad (4.4\text{a})\text{–}(4.4\text{i}).
\end{aligned}
$$

For simulation purposes, using the junction type discussed in Section 4.3.1, five vehicles are placed on junction arm_1 and six vehicles are placed on junction arm_2 with $N_{\text{p}} = 5$, $v_{\max} = 10\ \text{ms}^{-1}$ and $t_{\text{h}} = 1.7887$ s.

Results of this simplistic simulation are shown in Figure 4.7, which shows that, after an initial transient phase, the position and order of the vehicles settle to a stable interleaved configuration with minimal inter-vehicle interactions. Figure 4.6 overlaid vehicle trajectories from each collision set.

Remark: This collision set overlaying is possible because all vehicle and road geometries are non-changing and homogeneous; thus, their respective collision sets take the same shape and size, as a result, align together. Slight trajectory shifts appear (see Figure 4.6), mainly because the vehicles are reintroduced at the inlet at control synchronous times; thus, their inter-simulation time-step progress in position (the travelled distance within the buffer region) is reset. Furthermore, vehicles need to slightly modify their own control input to maintain the obstacle avoidance. For simplicity, the simulation time step is chosen identical to the control time step $\delta t = 0.5$ s.

In Figure 4.7, orderly traffic flow can be seen after the first junction crossing (transient phase). The flow stabilises to the interleaved crossing and each vehicle is close to the desired reference speed in the steady-state solution.

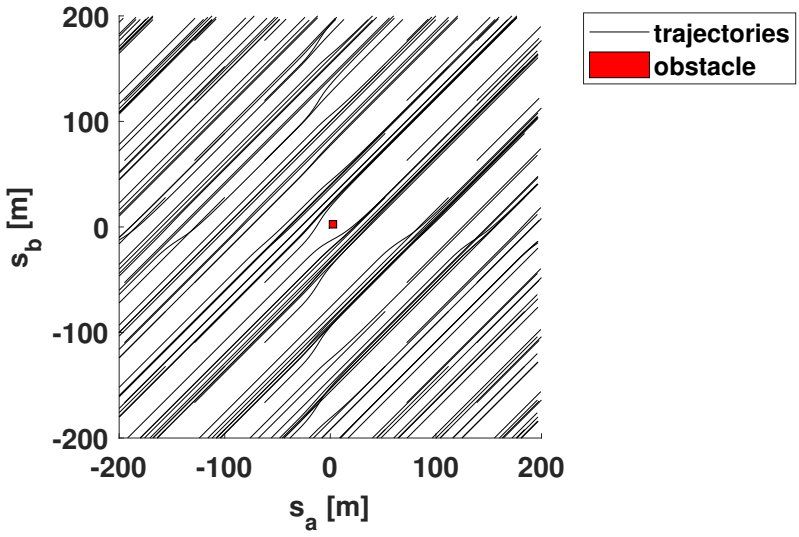


Figure 4.6: Overlaid vehicle trajectories for all collision sets $\mathcal{C}_{ab}$, $\forall\{a,b\} \in \hat{\mathcal{N}}_{\text{JCI}}$; all the trajectories are safely avoiding the cross-junction obstacle.

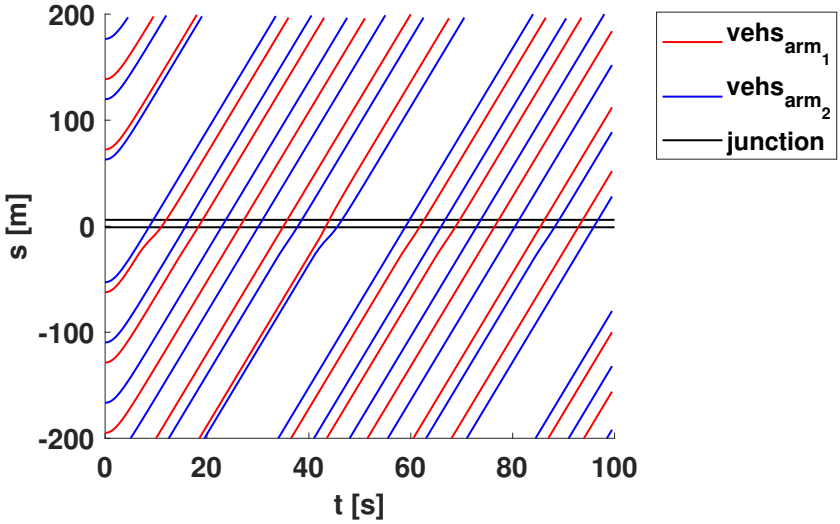


Figure 4.7: Space-time graph of vehicle trajectories on both junction arms; junction limit positions are indicated with solid black lines.

To introduce uncertainty and promote changes in the simulation (disturb the interleaved traffic), the initial v_{ref} speeds of each vehicle are drawn as uniformly distributed random variables from the range of $[6,\ 10]$ ms^{-1} when they appear or reappear at the inlet. Thus, compared to those in Figure 4.7, the trajectories in Figure 4.8 are shaped more dominantly by car-following and junction-crossing interactions as a result of the changing reference speeds. The trajectories in Figure 4.9 are significantly less orderly; however, around the magnified vicinity of the junction obstacle, the trajectories maintain a generous clearance. This clearance is closely related to the junction passing speed. Note that the constraint formulations are based on the time-headway parameter, which controls the speed-dependent safety distance from the obstacles.

In Figure 4.10, as expected from the clearance in Figure 4.9, the vehicle speeds dip around the junction but remain positive and above 1 ms^{-1}, meaning that no vehicle came

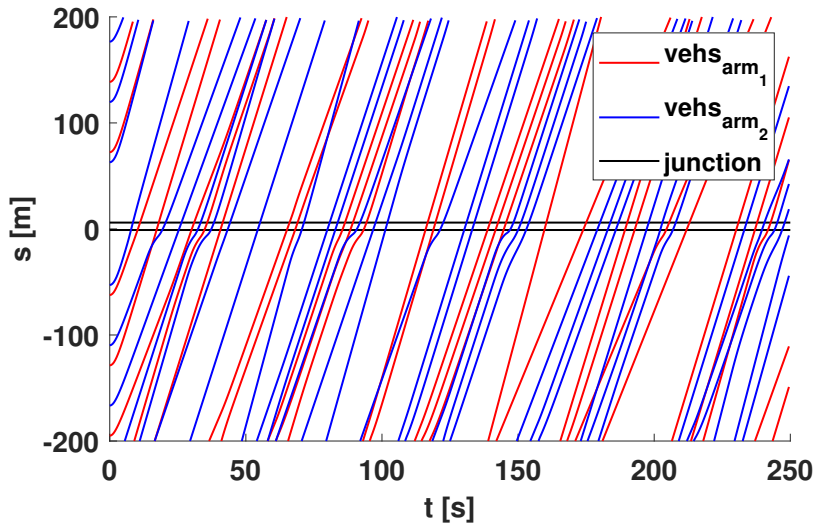


Figure 4.8: Space-time graph of vehicle trajectories on both junction arms; junction limit positions are indicated with solid black lines; vehicle reference speeds chosen in the range $[6, 10]$ ms^{-1}.

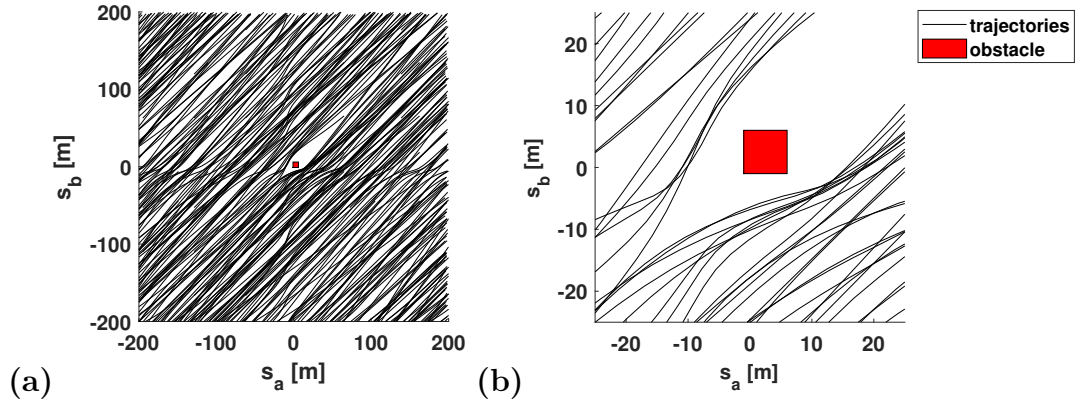


Figure 4.9: Vehicle trajectories overlaid and shown for all collision sets with the junction obstacle indicated in **(a)** and magnified in **(b)**.

to a full stop.

Two particular artefacts show the simulation issues of inlets and outlets, visible in Figure 4.10. Starting with the **outlet artefact**, the problem stems from the predictive nature of the control and the vehicles vanishing at the outlet. First, it must be understood that, as the reference speeds are random variables, the vehicles with higher reference speeds are likely to catch up with the vehicle travelling ahead in the same lane at a slower speed. After the faster vehicle catches up to the slower one, they settle on a mutually beneficial speed regulated by their car-following interaction and the related constraints. Once at the outlet, the slower vehicle is removed from the simulation, meaning that the vehicle behind it travelling slower than its desired speed can accelerate, since the car-following interaction with the slow leader vehicle is also removed. This speed-up artefact can be witnessed on some of the trajectories closest to the outlet.

The **inlet artefact** is also observed which, similarly, reflects the missing information of the traffic outside the simulation. Generally, when a control is turned on in a system,

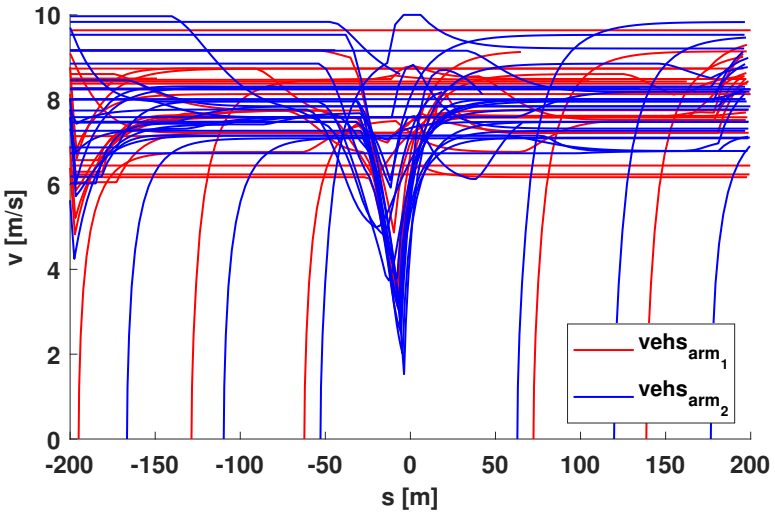


Figure 4.10: v–s graph of all vehicle trajectories on both junction arms showing inlet and outlet artefacts as well as the speed dip prior to the junction obstacle where vehicles give way.

it causes some control actions, because the states rarely match the desired or optimal states. The vehicles in the inlet appear with different reference speeds at times influenced by the outlet artefact through the synchronous reappearance policy. Their candidate initial speed is chosen as their new random reference speed. This candidate speed is generally distinct from the appropriate, desired, or optimal speeds in this situation. For example, two low-speed car-following vehicles leaving from the outlet and reappearing at the inlet as a leader with low speed but a follower with a higher reference speed would require heavy deceleration from the follower to respect safety and optimality of the control. In a real-world example, the traffic information may visually propagate upstream, resulting in feedback and causing the vehicle inflow to settle for a soft transition to the downstream traffic state. Thus, contrary to the simulation, real-world initial vehicle speeds and vehicle arrival times at a (virtual) inlet would already be affected by downstream states. Evidently, choosing inflow measures and parameters at random results in transient control state differences and requires extra control action (branded as inlet artefact).

Removing deactivated junction-crossing interaction constraints was discussed in Section 4.4. The computational speed-increase effect is shown in Figure 4.11; junction inlets were used to generate continuous streams of vehicles entering into the simulation and engaging in JCI (this junction type was discussed in Section 4.3.3). The controller is shown in MPC 4.2, in which the change from MPC 4.1 is the use of $\mathcal{N}_{\text{JCI}}$ over $\hat{\mathcal{N}}_{\text{JCI}}$. The optimisation speed results are shown in Figure 4.11 **(a)** while the number of junction-crossing interactions is shown in Figure 4.11 **(b)**. It is clear that after the first vehicles pass through the junction, the related junction-crossing interactions are arbitrated and can be removed, yielding an opening gap between the number of considered JCI and the direct effect on the optimisation time. The size impact on the optimisation problem is substantial due to the multiplicative nature of the number of collision sets (discussed in Section 4.4). However, the speed impact is not trivial, as the already arbitrated inter-

actions should yield low-complexity decision in the branch-and-bound method since the binaries are indirectly fixed by the vehicle states. This simple pre-processing results in faster calculation, exemplified by an average 33% decrease in computational time for this simulation. The total simulation time with all JCI included was 5.8 s for 500 control iterations (and, with only the active JCI, 3.86 s).

MPC-4.2

$$
\begin{aligned}
&J^{*}_{\text{MPC-4.2}} = \min\ (4.1) \\
&\text{s.t.} \quad \forall j \in \{0 \dots (N_{\text{p}} - 1)\},\ \forall n \in \mathcal{N}: \\
&\qquad\qquad x_n\,(k|k) = x_n\,(k) \\
&\qquad x_n\,(k+j+1|k) = A\ x_n\,(k+j|k) + B\ u_n\,(k+j|k) \\
&\qquad\qquad (x_n, u_n) \in \mathcal{X}_n \times \mathcal{U}_n \\
&\text{and } \forall (p,q) \in \mathcal{N}_{\text{CFI}}: \\
&\qquad (4.3) \\
&\text{and } \forall \{p,q\} \in \mathcal{N}_{\text{JCI}}: \\
&\qquad (4.4\text{a})\text{–}(4.4\text{i})
\end{aligned}
$$

Remark: The optimisation time was measured with the reasonably smooth Matlab tic-toc timer function because the built-in Gurobi timer was found to have crude discrete steps in the range of interest (milliseconds); it often returned 0 s for fast optimisations, which holds little value in a comparison. Additionally, the discontinuous and discrete steps of the Gurobi timer were found to bias statistical results compared to the closer-to-real-time high-resolution measurements from Matlab. In comparison, the Matlab tic-toc timer function, which was chosen for its consistency, incorporates an extra overhead of the solver interface time as well as other factors from the task scheduling of the operating system, resulting, in average, readings slightly higher than the raw optimisation time. The two simulation results (with MPC 4.1 and MPC 4.2) yield identical vehicle states, as expected, and, thus, are deemed to be a fair comparison.

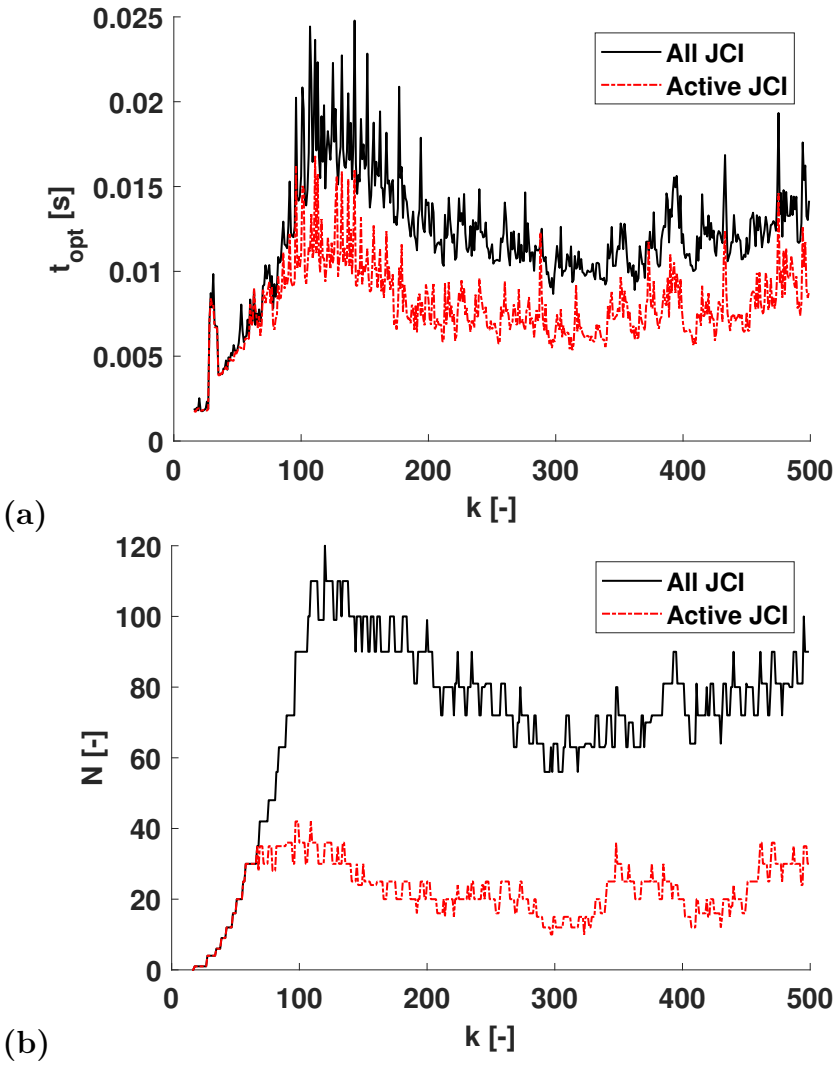


Figure 4.11: **(a)** Optimisation solve time; **(b)** number of all and active JCI.

4.5.2 FCFS fixed-order policy

FCFS junction passing is often held up in the literature as an example of a bad junction-passing policy [36]. Combined with the baseline control, it retains many benefits from the MPC formulation and lowers the decision-making computational cost, as the vehicle order is already fixed by the heuristic nature of the FCFS policy [43]. The performance of these fixed-order controls is attractive so long as the assumptions, heuristics and models hold well and no unexpected disturbances affect the states. Compared to the baseline control, these fixed-order controls trade computational complexity for vehicle-order adaptability. For example, a single vehicle slowing down due to a hazard on the road could warrant changes in the junction-crossing order but since this is fixed for the FCFS policy, it yields performance losses. The junction-crossing order in this FCFS example is generated from the initial order of the vehicles at the inlets. This implicitly removes the need for two hyperplanes; the only task is to switch from approaching the junction to passing through the junction. Thus, two of the four binaries and their respective hyperplanes can be

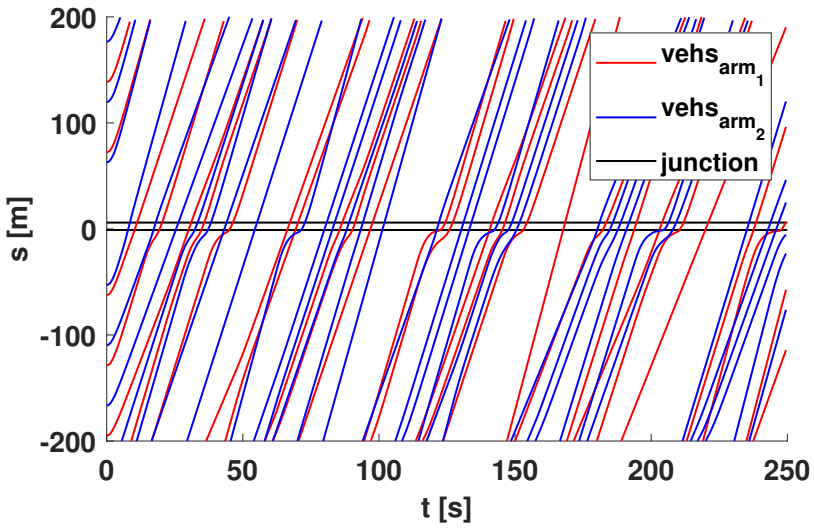


Figure 4.12: Trajectories for FCFS policy control with random vehicle reference speeds.

removed for each prediction time steps from (4.4). The separating hyperplane inequality also has to be amended for only two binaries. The two hyperplane inequalities and their time-shifted versions are in (4.8a)–(4.8d) with binary coupling in (4.8e).

$$s_p\left(k+j|k\right)+t_{\mathrm{h}}v_p\left(k+j|k\right)\leq-\frac{W}{2}+Mb_{1pq}(k+j|k) \tag{4.8a}$$

$$-s_q\left(k+j|k\right)\leq-\frac{W}{2}-L+Mb_{2pq}(k+j|k) \tag{4.8b}$$

$$s_p\left(k+j+1|k\right)+t_{\mathrm{h}}v_p\left(k+j+1|k\right)\leq-\frac{W}{2}+Mb_{1pq}(k+j|k) \tag{4.8c}$$

$$-s_q\left(k+j+1|k\right)\leq-\frac{W}{2}-L+Mb_{2pq}(k+j|k) \tag{4.8d}$$

$$\sum_{i=1}^{2}b_{ipq}(k+j|k)\leq 1, \tag{4.8e}$$

where the junction crossing interactions are now ordered (tuples) (i.e. $(p,q)\in\mathcal{N}_{\mathrm{JCI,FCFS}}$); the JCI for the FCFS policy are collected in $\mathcal{N}_{\mathrm{JCI,FCFS}}$ and fixed; the additional ordered vehicle pair indices are decided when a new vehicle enters the simulated junction. The control policy is shown in MPC 4.3. Additionally, note that an even more concise formulation could be made by simply defining $b_{2pq}(k+j|k)=1-b_{1pq}(k+j|k)$ and dropping (4.8e) from the formulation.

The simulation in Figure 4.12 shows traffic with the previously introduced inhomogeneous reference speeds. Model mismatch for the initial assumptions is present, for example, when a higher-speed vehicle appears later in the simulation compared to a lower-speed vehicle on the other arm but before the first vehicle crosses the junction. In this case, higher-speed vehicle starts loitering, as the vehicle-crossing order was decided without accounting for vehicle-specific travel speeds. This behaviour evidently lowers the efficiency of the junction control; however, it is simple to deploy and, thus, easy to calculate com-

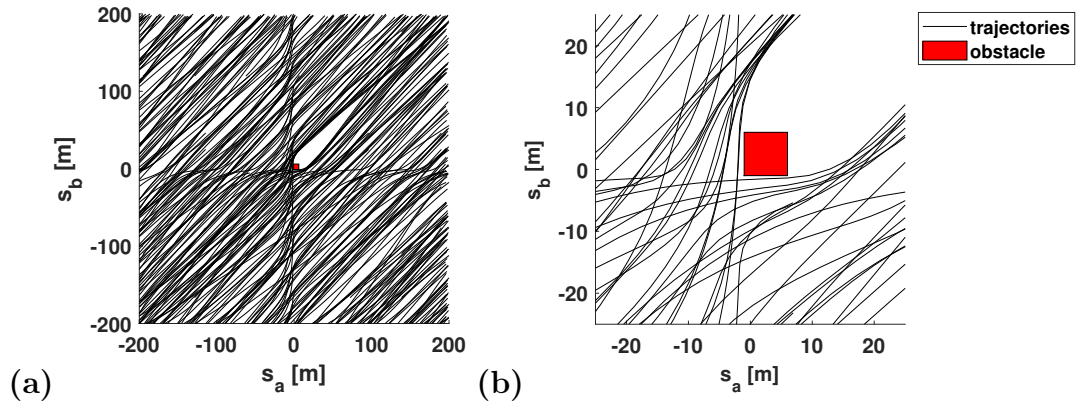


Figure 4.13: Vehicle trajectories overlaid for FCFS control and shown for all collision sets with the junction obstacle indicated in **(a)** and magnified in **(b)**

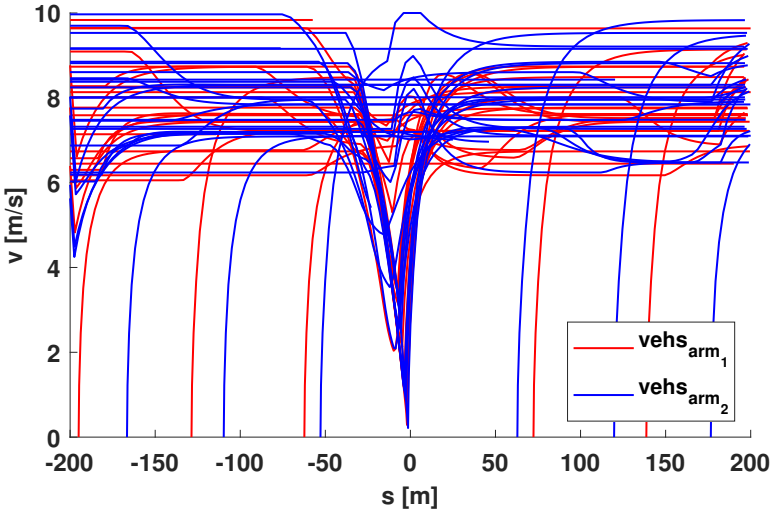


Figure 4.14: FCFS policy v–s graph of all vehicle trajectories on both junction arms, showing a deeper dip compared to the baseline policy results (Figure 4.10).

pared to incorporated full-decision-making in the baseline control. The trajectories on the overlaid collision-set planes (Figure 4.13) show a stronger slow-down effect (speed dip in Figure 4.14) in the vicinity of the junction. This is attributed to the cases where the vehicles that arrive at the junction early have to wait and lower their speed to respect to right of way of the other slower vehicles which have not arrived yet. Additionally, the vehicles passed the junction may freely change their order respect to the vehicles on the other arm (until the outlet position) driven by differences in v_{ref} desired speeds.

Figure 4.15 demonstrates the shortcoming of fixed-order policies. After a vehicle on arm_2 is artificially stopped, the vehicles arriving at the junction on arm_1 stop and wait as the policy demands that they cede the right of way even when they are clearly able to cross without interference. This example shows the policy-induced deadlock that can arise on account of the lack of adaptation in the control.

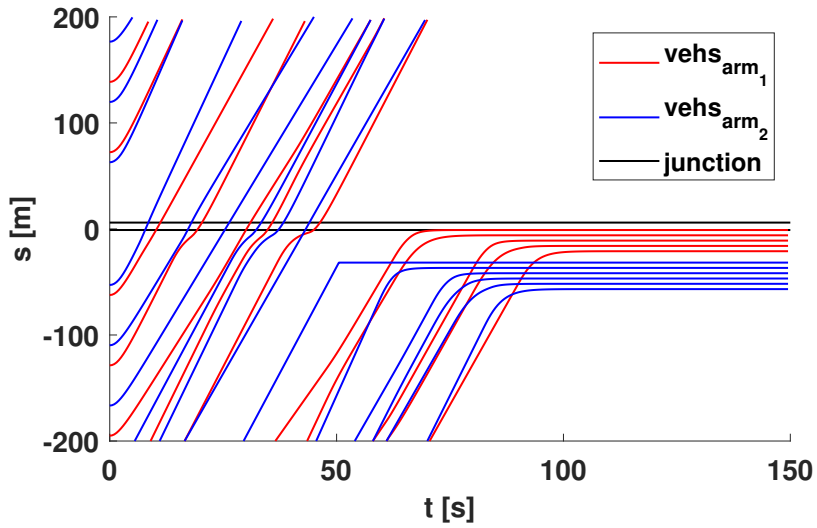


Figure 4.15: The vehicle on arm_2, which appeared at 30 s, has been stopped, halting all traffic on both arm_2 and arm_1 due to the fixed-order policy of FCFS.

MPC-4.3

$$J^*_{\text{MPC-4.3}} = \min\ (4.1)$$

s.t. $\forall j \in \{0 \dots (N_\text{p} - 1)\},\ \forall n \in \mathcal{N}$:

$$x_n\,(k|k) = x_n\,(k)$$

$$x_n\,(k+j+1|k) = A\ x_n\,(k+j|k) + B\ u_n\,(k+j|k)$$

$$(x_n, u_n) \in \mathcal{X}_n \times \mathcal{U}_n$$

and $\forall (p,q) \in \mathcal{N}_\text{CFI}$:

$$(4.3)$$

and $\forall (p,q) \in \mathcal{N}_\text{JCI,FCFS}$:

$$(4.8a)\text{–}(4.8e)$$

4.5.3 Baseline policy with soft pre-avoidance

In the case of merging, one advantage of the pre-avoidance shown in Section 3.5 is that a certain avoidance manoeuvre can be initiated earlier, as soon as the parameters of the obstructing collision set are available and considered in the formulation. The same can be done with box junctions by defining two new bounding hyperplanes enveloping the original obstacle from each side of the collision set; Figure 4.16 shows this proposed idea with a graphical interpretation.

By assuming a heuristic projection direction described by α, a projection view can be created that represents the conflicting states regarding the obstacle (see Figure 4.16). Essentially, the task is to softly evade the projected obstacle with the vehicle trajectories

(note that in the pre-merging formulation in Section 3.5, α was not explicitly considered and the concept of this direction was fixed there to $\alpha \equiv 1$). The α parameter, in this formulation, is chosen heuristically to the ratio of $v_{\text{ref}p}$ and $v_{\text{ref}q}$, where vehicles p and q engage in junction-crossing interactions. This heuristic may be a poor approximation when vehicles are unable to achieve their desired or reference speeds.

The cost function is constructed with soft penalties in the form of:

$$J = \sum_{n}^{\mathcal{N}} w_n \left(\sum_{j=0}^{N_\text{p}-1} \left[q\xi_n(k+j|k)^2 + ru_n(k+j|k)^2 \right] + q_\text{f}\xi_n(k+N_\text{p}|k)^2 \right) + \sum_{\{a,b\}}^{\mathcal{N}_\text{JCI}} \sigma_a F_{\text{sep } ab,a} + \sigma_b F_{\text{sep } ab,b}, \tag{4.10}$$

where the soft cost penalty terms are $F_{\text{sep } ab,a}$ and $F_{\text{sep } ab,b}$ with their relative weights σ_a and σ_b, respectively; vehicle indices are $\{a, b\} \in \mathcal{N}_\text{JCI}$.

The related soft constraints are:

$$s_a\,(k+N_\text{p}-1|k) + t_\text{h} v_a\,(k+N_\text{p}-1|k) - F_{\text{sep } ab,a} \leq \frac{v_{\text{ref } a}}{v_{\text{ref } b}} \left(s_b\,(k+N_\text{p}|k) - \frac{W}{2} - L \right) - \frac{W}{2} + M b_{ab,1}(k+N_\text{p}|k) \tag{4.11a}$$

$$s_b\,(k+N_\text{p}-1|k) + t_\text{h} v_b\,(k+N_\text{p}-1|k) - F_{\text{sep } ab,b} \leq \frac{v_{\text{ref } b}}{v_{\text{ref } a}} \left(s_a\,(k+N_\text{p}|k) - \frac{W}{2} - L \right) - \frac{W}{2} + M b_{ab,3}(k+N_\text{p}|k) \tag{4.11b}$$

$$-F_{\text{sep } ab,a} \leq 0 \tag{4.11c}$$

$$-F_{\text{sep } ab,b} \leq 0, \tag{4.11d}$$

where the vehicle box length is L and width is W; the terminal relaxation binary $b_{ab,1}$ is identical to the binary used in (4.4a) and the terminal relaxation binary $b_{ab,3}$ is identical to the binary used in (4.4c) box-junction obstacle avoidance. The complete control framework with the amended baseline control is shown in MPC 4.4.

The formulation builds on the weighted error between the soft avoidance hyperplane and the terminal states of the vehicle pair in the collision set, while it supplements the existing JCI from (4.4a)–(4.4i).

Simple numerical examples are shown for two vehicles in Figure 4.17, where the initial position states are in conflict with the projected obstacle view. The vehicles immediately start their corrective manoeuvre to move out of the set of states, that later would yield to the need for avoidance actions. This may be beneficial, as vehicles can softly sort out their avoidance before reaching the vicinity of the junction, in which the avoidance constraints are hard and more costly doing them later than sooner (e.g. intensive accelerations and decelerations). The controller used in this simulation is summarized in MPC 4.4.

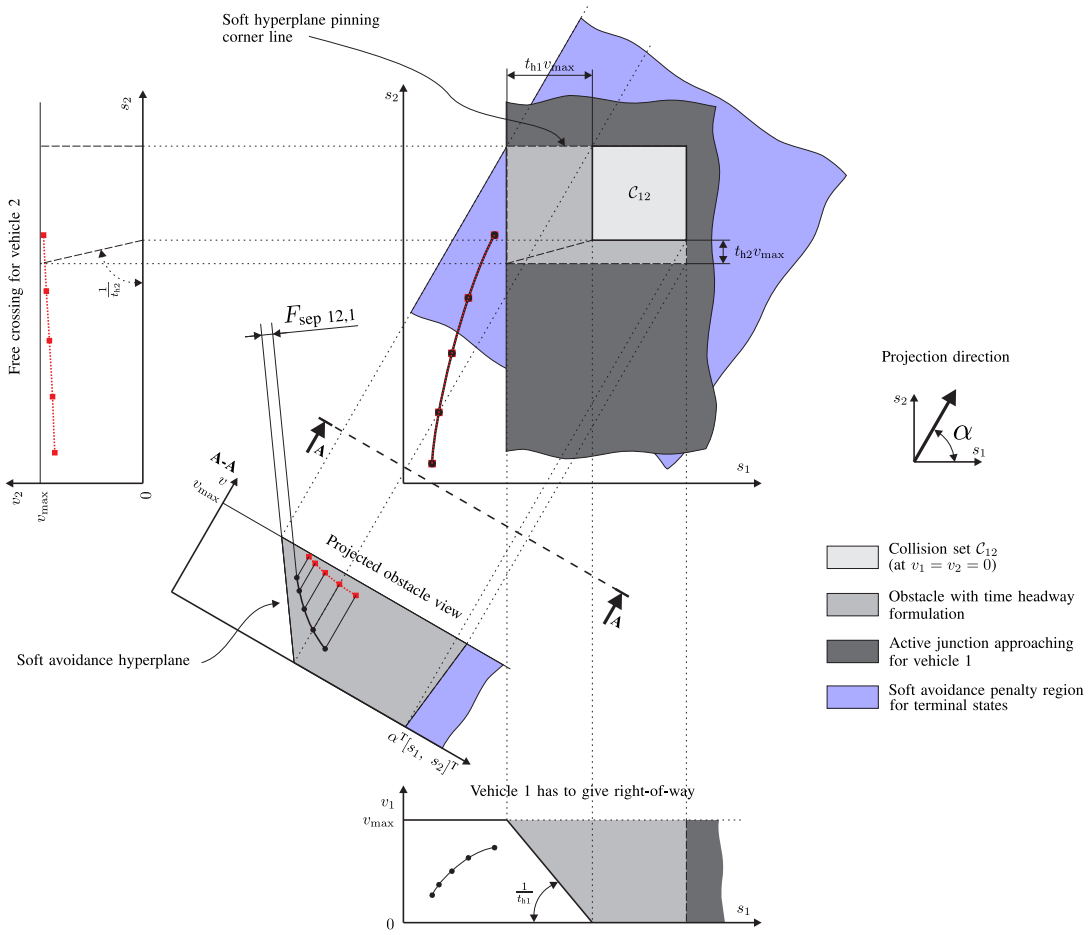


Figure 4.16: Methodology of soft pre-avoidance for cross junctions, where an additional soft constraint is fitted (for both leader-follower cases) with the additional directional information of α. This soft cost violation provides the force to move the system trajectories outward from the projection of the junction obstacle conflict states.

MPC-4.4

$$J^*_{\text{MPC-4.4}} = \min\ (4.10)$$

s.t. $\quad \forall j \in \{0 \dots (N_\text{p} - 1)\},\ \forall n \in \mathcal{N}$:

$$x_n\,(k|k) = x_n\,(k)$$

$$x_n\,(k+j+1|k) = A\ x_n\,(k+j|k) + B\ u_n\,(k+j|k)$$

$$(x_n, u_n) \in \mathcal{X}_n \times \mathcal{U}_n$$

and $\forall (p,q) \in \mathcal{N}_\text{CFI}$:

$$(4.3)$$

and $\forall \{p,q\} \in \mathcal{N}_\text{JCI}$:

$$(4.4\text{a})\text{–}(4.4\text{i}),$$

$$(4.11\text{a})\text{–}(4.11\text{d}), \qquad \text{with } a \equiv p,\ b \equiv q$$

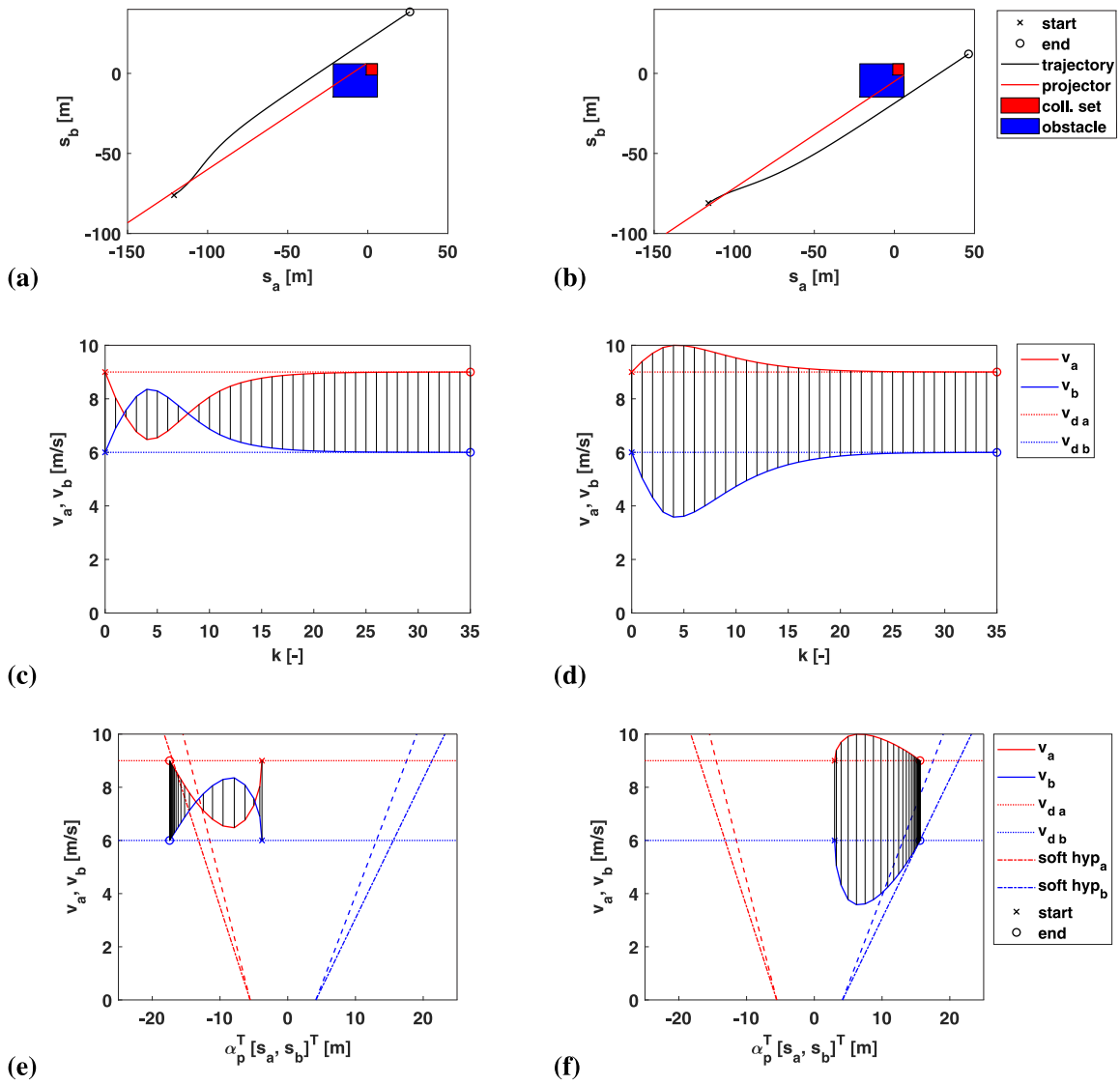


Figure 4.17: Soft pre-avoidance simulation trajectories for s_b-leader case in **(a)**, **(c)** and **(e)** and the s_a-leader case in **(b)**, **(d)** and **(f)**; **(a)** and **(b)** graph the obstacle dimensions that were indicated for the desired speeds of the agents; **(e)** and **(f)** show the trajectories on the projected view.

Figure 4.18 depicts a related experiment run for a multiple-vehicle simulation, in which the trajectories indicate more crossing vehicles establish nearly ideal gap clearance before reaching the junction. The same tendency is clear in Figure 4.19, in which the clearance around the junction obstacle is much more definitive than that in the baseline (Figure 4.9), and FCFS (Figure 4.13) simulations. This can be advantageous because it does away with most need for intensive acceleration and deceleration in and near the junction, enabling vehicles to pass with little arbitration and at higher speeds. In real-world cases, the decision-making is a hard and cognitively intensive process for human drivers, which could be a factor behind accidents in junctions; however, the soft pre-avoidance allows for earlier decisions by leaving trivial choices available by the time the vehicle arrives at the junction. Figure 4.20 shows the trajectories and the mark of control actions of soft

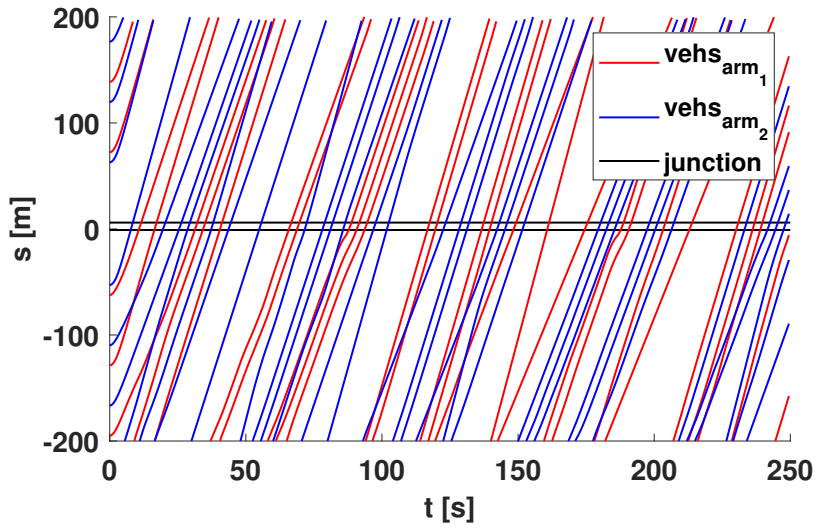


Figure 4.18: Space-time graph of soft pre-avoidance-policy-controlled vehicle trajectories on both junction arms; the vehicle reference speeds are chosen in the range $[6; 10]$ ms^{-1}.

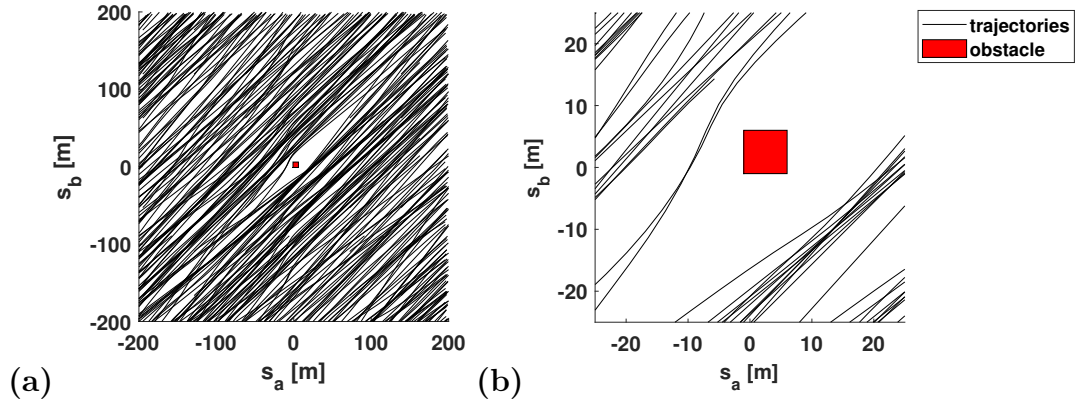


Figure 4.19: Overlaid soft pre-avoidance-policy-controlled vehicle trajectories for all collision sets with the junction obstacle indicated in **(a)** and magnified in **(b)**.

pre-avoidance policy; it depicts vehicles modifying their controls and slightly adjusting their speed to open an even clearance gap before the junction is reached.

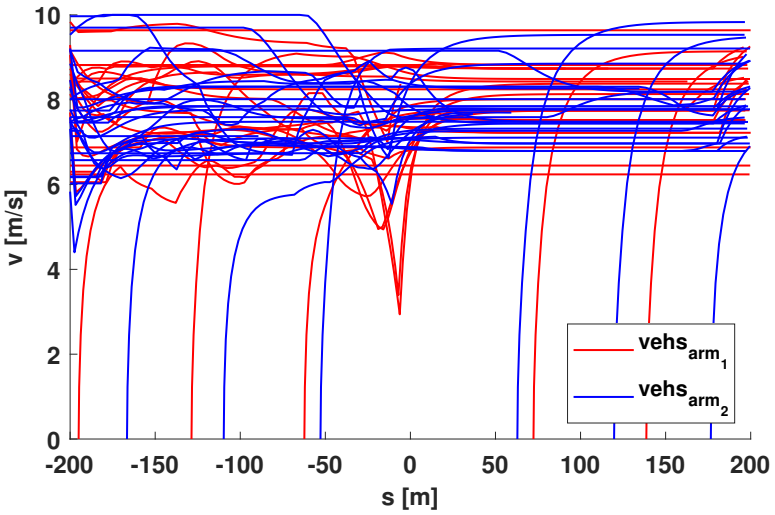


Figure 4.20: Soft pre-avoidance policy v–s graph of all vehicle trajectories on both junction arms showing an increased control action before the junction, resulting in minimal speed dip and less intensive accelerations and decelerations at the crossing; for comparison see the baseline (Figure 4.10) and FCFS (Figure 4.14) simulation results.

4.6 Fundamental diagram and deadlocks

Deadlocks or gridlocks in traffic systems may arise from the structural characteristics of roadways or decision-making issues. Some of the issues may stem from cyclic priority graphs, as argued by [31]. Other sources for deadlocks are related to decision-making (e.g. control policies, decision variable types and discrete logic) with model and solver choices (e.g. approximate solutions through heuristics and rule based behaviours) and limited framework capabilities (e.g. requirements for non-negative vehicle speeds and operational limits). In case of decentralised optimisations, Nash-equilibrium is another source of deadlocks.

In this section, the fundamental diagram [75] will be utilised with 8-loop junction scenarios (see Figure 4.3), to analyse traffic measures for the control policies. The 8-loop junction also serves as a good example of deadlock scenarios in crowded systems, which brings to mind the notion of box junctions to prevent these situations.

As shown in Figure 4.3, the 8-loop junction has the characteristic length d_c; thus, a full circle on the loop is $4d_c$. The density of vehicles on a strip of road is as follows:

$$\rho = \frac{N}{d}, \tag{4.13}$$

where the number of vehicles is N over a road strip of length d. Furthermore, the traffic flow is as follows:

$$Q = \frac{N}{\Delta t} = \rho \bar{v}, \tag{4.14}$$

where the number of vehicles is N, which pass through a road position over the time span

of Δt. Alternatively, the flow can be calculated with the use of average flow speed $\bar{v}$ and traffic density ρ. The flow measure Q is usually expressed in veh/h while the density ρ is usually expressed in veh/km.

In the simulation examples, the number of vehicles (agents) is fixed to $N = 10$; by changing d_c, the desired average simulation density ρ can be set.

$$\rho(d_c) = \frac{N}{4d_c} \tag{4.15}$$

For a single road strip, the critical flow $Q_{cr,s}$ can be found by assuming that all vehicles travel at the homogeneous desired speed (reference speed) v_d and calculating the $\rho_{cr,s}$. Vehicles, on average, require the desired space $L + (t_\mathrm{h} + dt)v_d$, which covers for vehicle length, time headway and spatio-temporal leader-follower spacing, defined by (4.3). Thus, without any additional gap:

$$\rho_{cr,s} = \frac{1}{L + (t_\mathrm{h} + dt)v_d} \tag{4.16}$$

yielding:

$$Q_{cr,s} = \frac{v_d}{L + (t_\mathrm{h} + dt)v_d}. \tag{4.17}$$

The highest density possible on a single road strip requires vehicles to have zero speed; thus:

$$\rho_{\max,s} = \frac{1}{L} \tag{4.18}$$

and, thus, the flow would be $Q = 0$. Within the simulation, the vehicles generally change their speed; for this reason, Q average traffic flow is calculated with:

$$Q = \frac{\sum_{n\in\mathcal{N}} \Delta s_n}{4d_c\Delta t}, \tag{4.19}$$

where the cumulated distance is Δs_n, travelled by vehicle n over simulation time Δt.

4.6.1 Numerical experiments on the 8-loop junction

In the experiments, the simulation time Δt is chosen to ensure that vehicles can travel long enough to get past the initial transient flows. When possible, this means that vehicles drive multiple circles over the track or for a given time, when the track is congested, in order to exhibit representative measures of non-transient processes (e.g. average Q flow). Five laps are chosen as a guideline for simulation length, assuming desired speed as the average travel speed for a specific d_c; however, simulation length should not fall below 150 s. This minimal time limit leaves time for traffic to settle in cases of high density

traffic simulations, in which whole laps cannot be completed in a reasonable amount of time due to slow speeds and longer lap times.

With the vehicle width $W = 2$ m and, gross length $L = 5$ m ($\hat{L} = 4$ m plus added safety length), $\rho_{\max,s} = 200$ veh/km. Moreover, $\delta t = 0.5$ s, $t_\mathrm{h} \approx 1.7887$ s and $v_d = 8$ ms^{-1}; the critical flow for the single lane is $Q_{cr,s} \approx 1236$ veh/h at $\rho_{cr,s} \approx 42.9$ veh/km. Initial vehicle positions are shown in Figure 4.21 **(a)** keeping an equal gap between the vehicles as well as between the first vehicles and the entry point of the junction. Moreover, a vehicle-worth of space is left unoccupied following the junction to allow the first vehicles to pass through without becoming an obstruction. In Figure 4.21 **(b)**, the limit-density constellation is indicated when one of the loops could end up with all of the vehicles, resulting in the existence of deadlocks in this 8-loop system at this density or any higher densities. The limit density can be calculated as $\rho_{lim} = \dfrac{N}{2(NL+W)} \approx 96.15$ veh/km because the setup is symmetrical and, thus, the loop lengths are equal. An initial constellation of the test simulation with the highest density is shown in Figure 4.21 **(c)**, still keeping the junction and the following region free at $\rho = \dfrac{N}{NL+2L+2W} \approx 156.3$ veh/km.

Simulation results are shown in Figure 4.22 for horizon lengths of $N_p = \{1,\ 3,\ 10\}$, starting from zero initial vehicle speed and calculated with the baseline control MPC 4.2, which was formulated for the 8-loop junction (with added extra car-following constraints). As expected, vehicles are able to reach and keep their desired speed v_d in the low-density simulations because of the relatively low rate and severity of inter-vehicle actions (e.g. junction crossing interactions).

In a simple hypothesis case, the same, Q average 8-loop-system flow is hypothesised on the two junction-inbound road sections. Furthermore, the cross junction would experience the superposition of its junction arm flows (the junction bandwidth would be $Q_\mathrm{junc} = 2Q$ flow in an ideal case). Moreover, the critical flow $Q_{cr,s}$ was already derived and is expected to be the maximum average single-lane flow. However, in a trivial approach, the bandwidth of the junction is double the incoming road flow. This is expected to define the upper bound on the average system flow and flows in the fundamental diagram, thus, $Q \leq Q_{cr,s}/2$. However, in the simulations, consistently higher-than-expected flows were observed for mid-range traffic densities (e.g. in Figure 4.22). This can be explained by the non-homogeneous and non-symmetrical junction-inbound flows, as it is more efficient to let a group of vehicles (platoon) through a junction—since they only have car-following interaction (CFI) among themselves—than it is to alternate junction arms with vehicles sharing junction-crossing interactions (JCI). If two vehicles cross the junction on the same road (i.e. with CFI between them) they would occupy $2L$ physical length if the speed-dependent safety distance is disregarded for a simplistic comparison (or if the vehicle speeds are zero). Then, in the case where two vehicles have JCI between them, they cross the junction on separate roads and occupy at a point in time at least $2L + W$ physical space on the roads (once more, with the speed-dependent safety distance disregarded).

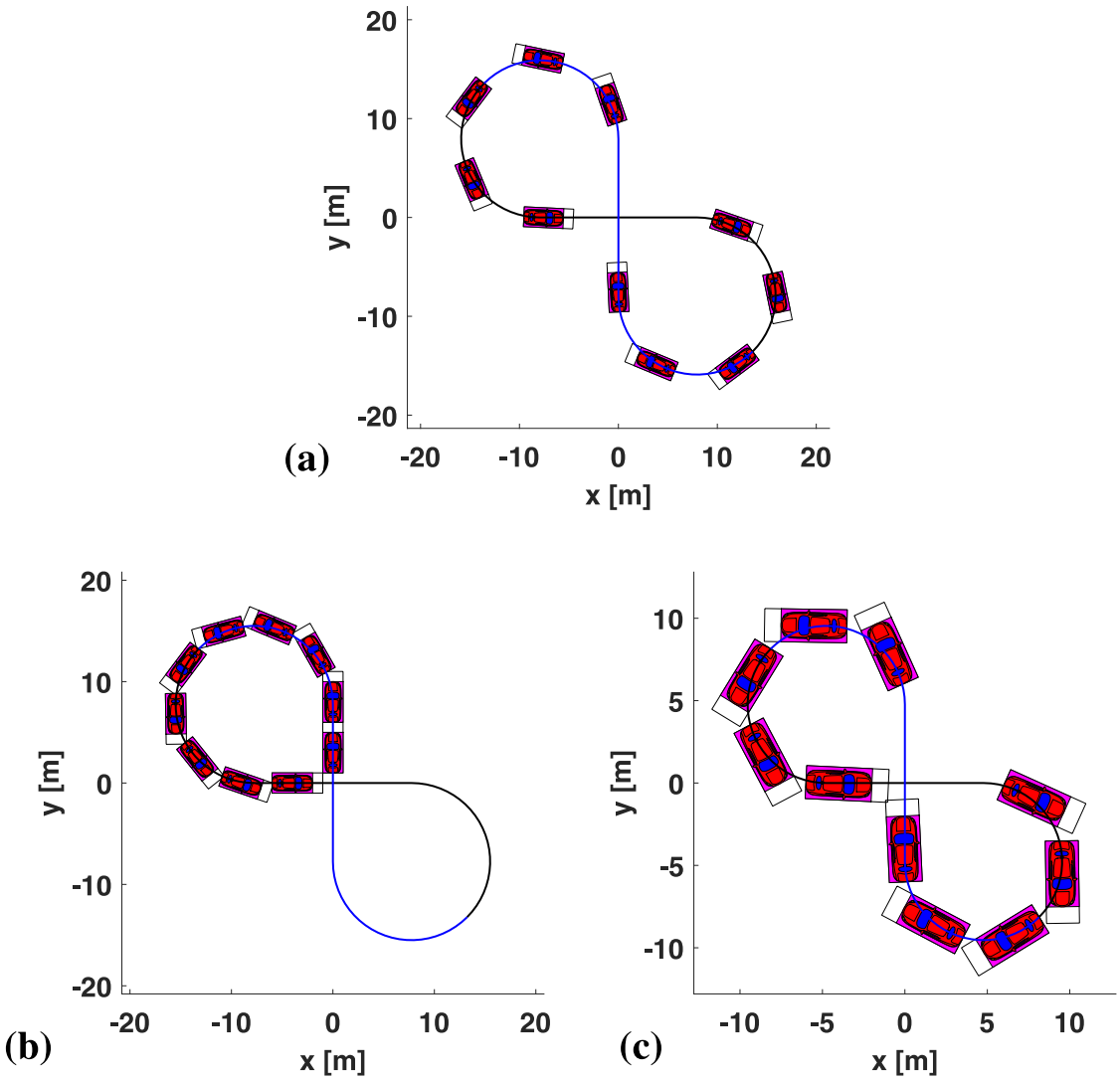


Figure 4.21: **(a)** initial vehicle distribution in the simulation; **(b)** existence of deadlock configuration at $\rho_{lim} \approx 96.15$ veh/km; **(c)** initial vehicle distribution in the final simulations chosen $\rho \approx 156.3$ veh/km as the highest density.

This difference offers a slight advantage to clustering vehicles on the same road to cross the junction, as in the case with platoons and traffic lights (signalised junctions). Clustering several vehicles generally requires some form of coordination with extra acceleration or deceleration necessary to achieve and maintain the desired inter-vehicle gaps. This extra effort is likely to be comparatively disadvantageous in terms of performance, as the control actions are penalised in the cost function.

Moreover, vehicles travelling on the 8-loop track form, higher-, and lower-than-average density sections and even different speeds (e.g. likely, high, desired speeds where the vehicle experiences a low-density (or free-flow) region). This is evident when the two single loops of the 8-loop track are examined as separate vehicle containers; two vehicle exchanges between the loops cannot transpire simultaneously, as only one vehicle may occupy the junction conflict zone at a time. Consider a non-deadlock case in which vehicles flow from one loop into the other, and vice versa, concluding that while the number of

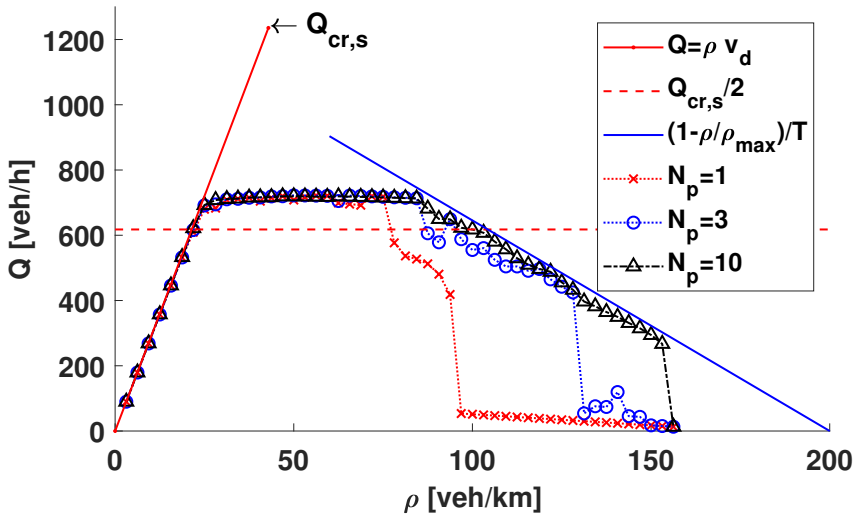


Figure 4.22: Fundamental diagram of average vehicle flows in the 8-loop-junction simulations for $N_\mathrm{p} = \{1,\ 3,\ 10\}$ horizon lengths; blue solid line serves as an empirical upper bound on flows with $T = t_\mathrm{h} + 2\delta t$.

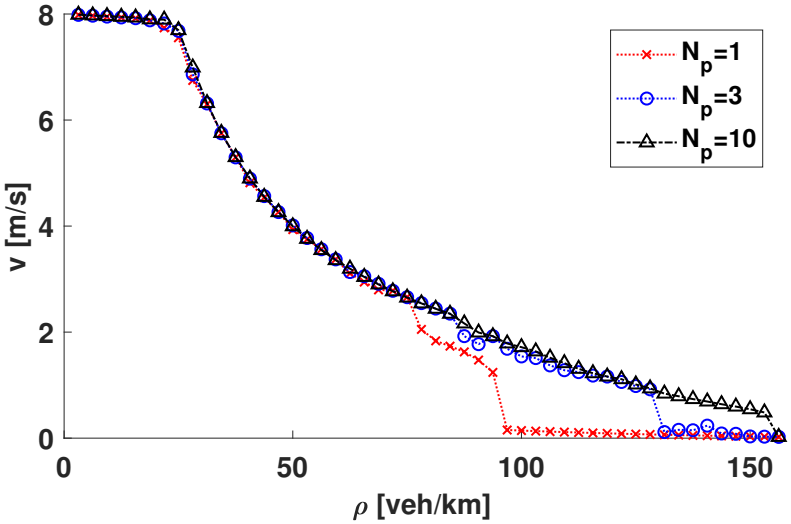


Figure 4.23: Average speeds of the vehicle flows in the 8-loop-junction simulations for $N_\mathrm{p} = \{1,\ 3,\ 10\}$ horizon lengths.

vehicles is changing on each of the two loops, the loop lengths remain constant, yielding an inherent change in traffic density. Thus, the density is, generally, not fixed between different parts of the track, it changes and fluctuates respect time and position.

Another density-related phenomenon is the formation of vehicle queues with higher local-traffic density prior to the junction entry point (or bottleneck in the system). However, this effect is light, as there are only a few vehicles in the simulated system, due to the complexity limitations of the MI-MPC.

Recall, that the simulation results reached higher average flows (Figure 4.23) than it was expected by the limiting upper bound on junction throughput (in the hypothesised symmetric flow case $Q \leq Q_{cr,s}/2$). This is also explained as the effect of the centralised optimal control calculating globally optimised vehicle velocity profiles, in which the

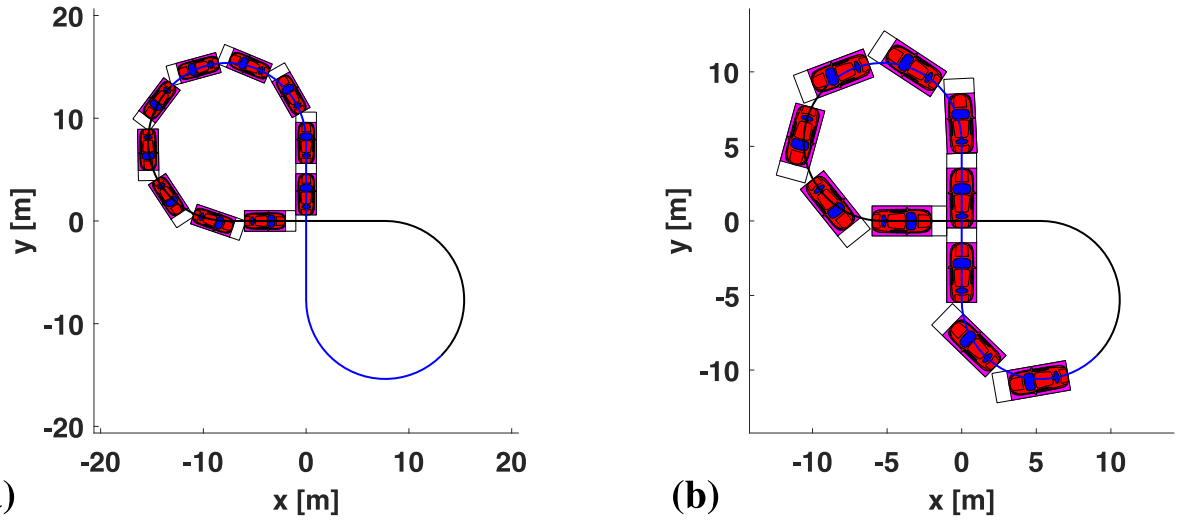


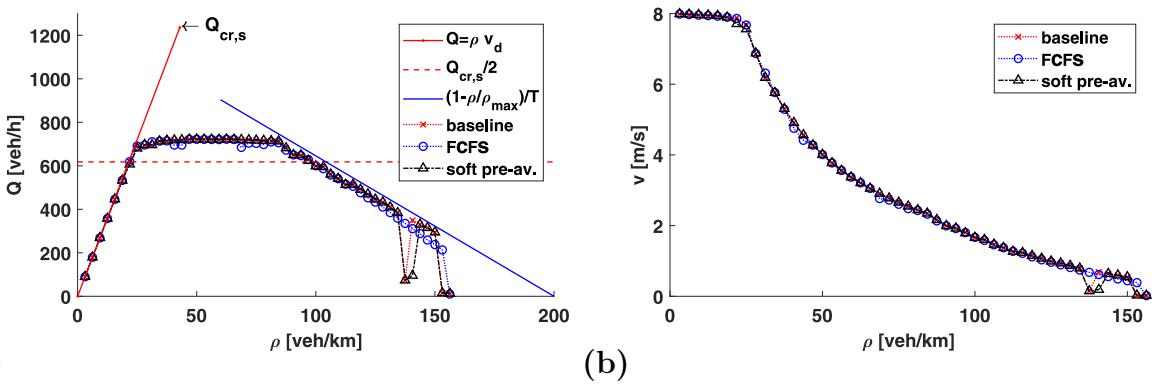
Figure 4.24: **(a)** First possible deadlock configuration at $\rho \approx 96.87$ veh/km for $N_\mathrm{p} = 1$; **(b)** deadlock configuration at $\rho \approx 140$ veh/km.

optimiser is able to exploit local density differences, non-symmetric and non-stationary flows to reach more efficient higher average flows than the hypothesised limit.

In the high-traffic-density region, the vehicles can only maintain lower-average speeds due to the high interactions between them and the low availability of inter-vehicle gaps. The interesting fact is that, over the ρ_{lim} limit density, the simulation may end up in a deadlock situation; however, for longer prediction horizons the simulations remained operational above the limit density. This shows the resilience of predictive control against simple trivial deadlocks.

In Figure 4.22, the lower Q flow values in the high-density region correspond to the deadlock situations; these flows are non-zero, because between the initial configuration and the deadlock configurations the vehicles cover a non-zero distance, resulting in non-zero average measures. These deadlock situation, however, easily identifiable from the average measures because they discontinuously jump to a low value (e.g. in the fundamental diagrams). Figure 4.23 shows the average flow speeds for the three horizon lengths $N_p = \{1,\ 3,\ 10\}$, reaching the desired speed at the low-density simulations and reaching near-zero speed at the simulations experiencing the high-density deadlock configurations. For $N_\mathrm{p} = 1$, Figure 4.24 **(a)** shows the deadlock configuration at $\rho \approx 96.87$ veh/km, which is in line with $\rho > \rho_{lim}$. Furthermore, this shows that the vehicles with short predictive capabilities are more prone to end up in deadlock situations. Figure 4.24 **(b)** shows a deadlock in a denser traffic simulation at $\rho \approx 140$ veh/km which would require more foresight and fairness between vehicles to prevent.

As discussed in Section 4.3.2, the 8-loop road structure is formulated on the straight road core simulation, with two extra car-following constraints and visualised as a projection to the 8-loop (see Figure 4.3). To test the control policies from Section 4.5, simply the extra car-following constraints need to be added which does not change the structure of the controller. Thus, the FCFS MPC 4.3 and soft pre-avoidance MPC 4.4 policies are



(a) (b)

Figure 4.25: Baseline, FCFS, and soft pre-avoidance 8-loop-junction simulations for $N_\mathrm{p} = 5$ horizon length; **(a)** fundamental diagram; **(b)** average vehicle flow speeds; blue solid line is serves as an empirical upper bound on flows with $T = t_\mathrm{h} + 2\delta t$.

compatible with the 8-loop simulation type similarly as the previous baseline example with MPC 4.2. The FCFS crossing orders (priorities) are decided at d_c characteristic distance from the centre of the junction same as in Section 4.5.2. The average flow and flow speeds, and densities are shown in Figure 4.25 with $N_\mathrm{p} = 5$ horizon length to enable comparison between the baseline, FCFS, and soft pre-avoidance policy simulations. All three policies largely performed similarly, though the FCFS did sometimes achieve lower operational flows than the other two controls. However, FCFS is the only policy which did not result in a deadlock breakdown until reaching the highest density scenario (which was explicitly designed to result in deadlock configuration). This resilience and the lower flows come from the fact that it heuristically fixes vehicle-crossing order in advance of the junction; the inferior adaptability of the FCFS controller in terms of crossing order ensures that bad traffic choices that lead to deadlocks and could not be forecast by the control are not made.

4.6.2 Passing completion in the box junction

Box junctions are popular in areas where traffic is highly susceptible to deadlocks, gridlocks due to local traffic specific properties (e.g. due to low visibility, high density, or asymmetric traffic inflows). There are two main approaches: impose a minimal speed across the junction, as discussed in [3] and formulated with time-headway safety in Section 3.6; or demand the vehicle to be in front of or past the crossing area in the foreseeable future, practically implementing a hysteresis switch [18]. According to de Campos et al. [18], the problem is manually split into two sub-problems seeking the better option to either cross or not cross the conflicting road section when another vehicle also aims to do the same; the considered vehicle orders were a heuristic subset of all possible crossing orders in the decision-making problem. Furthermore, each sub-problem-calculation sequence is repeated in the usual receding-horizon manner of MPCs. The minimal speed requirement seen in [3] does not provide the emergency stop capability in the region.

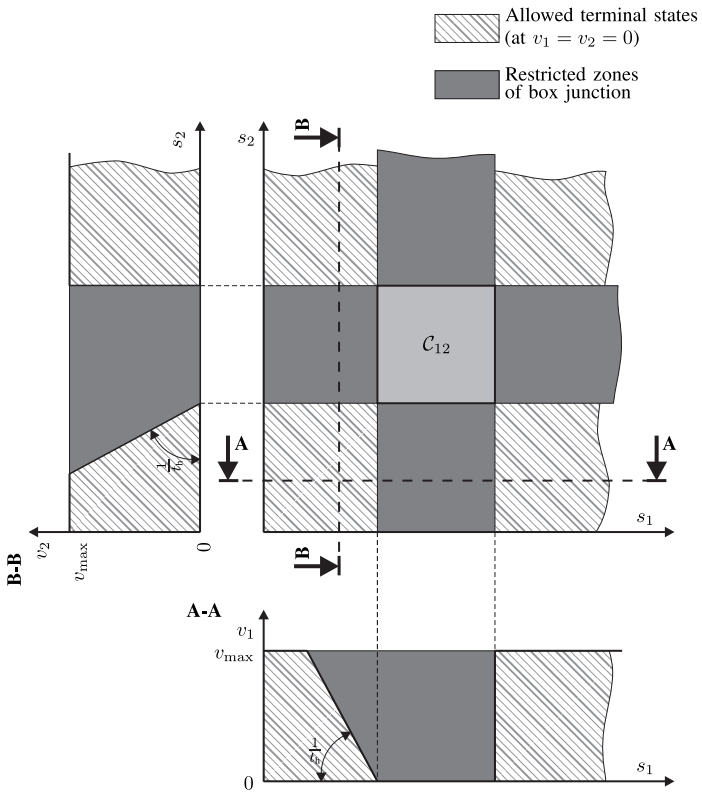


Figure 4.26: Terminal state sets for junction passing completion.

Using this approach with the finite-horizon formulation would also allow for the entry of vehicles that cannot see far enough into the intersection to determine if they can safely leave this mandatory-minimal-speed region. A vehicle may enter so long as it can keep its speed higher than the minimal speed limit. However, when this region is long and unseen (spans over the end of the horizon) and an exception may rise mid-way (e.g. when an obstruction is encountered ahead of the vehicle), then the vehicle would not be able to slow down and stop while abiding the speed limit. A safety example is phrased as 'never proceed into an intersection if someone else has the right of way' in [78]; as noted there, this rule can be formulated as linear-temporal safety logic.

This section introduces the hysteresis effect on the terminal vehicle states within collision sets of junction-crossing interactions. Where the allowed terminal states exclude the states, that would let the vehicle to be present, at this prediction time, in the junction (conflicting road section); such a terminal set is shown in Figure 4.26. It is clear that the same hyperplanes are defining the forbidden terminal states as the ones used for junction-crossing interactions but with different binding logic.

In Figure 4.26, all of the four allowed-terminal-state sets are defined and can be selected when the right two hyperplanes from the four are activated (the hyperplane choices are

handled inherently by the relaxation), thus:

$$s_p\left(k+N_\mathrm{p}|k\right)+t_\mathrm{h}v_p\left(k+N_\mathrm{p}|k\right)\leq-\frac{W}{2}+Mb_{pq,1}(k+N_\mathrm{p}|k) \tag{4.20a}$$

$$-s_p\left(k+N_\mathrm{p}|k\right)\leq-\frac{W}{2}-L+Mb_{pq,2}(k+N_\mathrm{p}|k) \tag{4.20b}$$

$$s_q\left(k+N_\mathrm{p}|k\right)+t_\mathrm{h}v_q\left(k+N_\mathrm{p}|k\right)\leq-\frac{W}{2}+Mb_{pq,3}(k+N_\mathrm{p}|k) \tag{4.20c}$$

$$-s_q\left(k+N_\mathrm{p}|k\right)\leq-\frac{W}{2}-L+Mb_{pq,4}(k+N_\mathrm{p}|k) \tag{4.20d}$$

$$\sum_{i=1}^{4}b_{pq,i}(k+N_\mathrm{p}|k)\leq 2, \tag{4.20e}$$

where the constraints are formulated $\forall\{p,q\}\in\mathcal{N}_\mathrm{JCI}$. This implemented hysteresis can be added to the discussed controllers, for example, the baseline vehicle-control formulation with the above, (4.20), terminal hysteresis constraints is shown in MPC 4.5.

MPC-4.5

$J^*_{\text{MPC-4.5}}=\min$ (4.1)

s.t. $\forall j\in\{0\ldots(N_\mathrm{p}-1)\},\ \forall n\in\mathcal{N}$:

$$x_n\left(k|k\right)=x_n\left(k\right)$$

$$x_n\left(k+j+1|k\right)=A\ x_n\left(k+j|k\right)+B\ u_n\left(k+j|k\right)$$

$$(x_n,u_n)\in\mathcal{X}_n\times\mathcal{U}_n$$

and $\forall(p,q)\in\mathcal{N}_\mathrm{CFI}$:

(4.3)

and $\forall\{p,q\}\in\mathcal{N}_\mathrm{JCI}$:

(4.4a)–(4.4i),

(4.20a)–(4.20e).

This formulation has the notion of the minimal horizon length that is necessary for a vehicle to cover the space between these terminal hyperplanes, or to clear the conflicting junction zone. Trivially short horizon lengths (in time) would allow only a short distance to be covered, which is insufficient for the terminal states to cross the junction. The switching mechanism would rather choose to not enter the junction in these cases. This effect can be seen in Figure 4.27 and Figure 4.28, where the baseline control with passing completion feature (MPC 4.5) cannot cross the junction with $N_\mathrm{p}=3$ horizon length. However, for the $N_\mathrm{p}=6$ and $N_\mathrm{p}=10$ cases, the proposed control scheme has a sufficiently long horizon length to make crossing possible and allow for safe crossing manoeuvres with $\delta t=0.5$ s. Furthermore, vehicles must verify not only that they are able to cross a junction

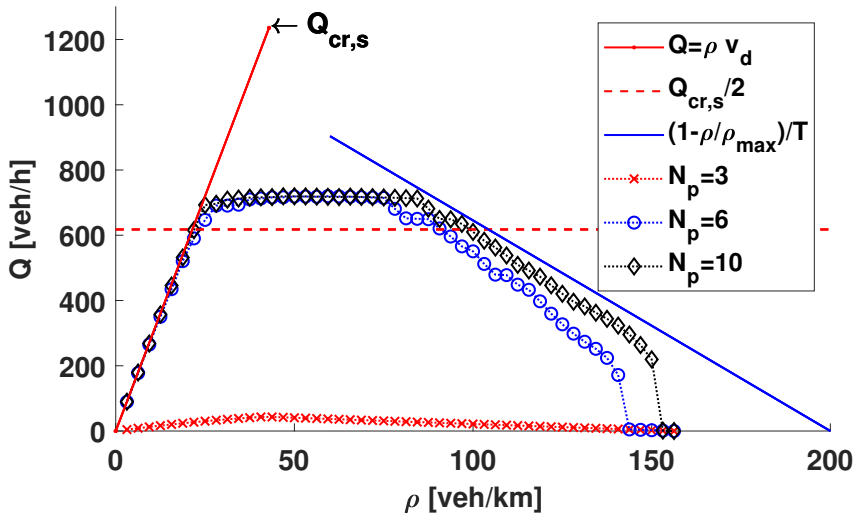


Figure 4.27: Fundamental diagram with baseline policy and added passing completion with $N_\mathrm{p} = \{3,\ 6,\ 10\}$ horizon lengths.

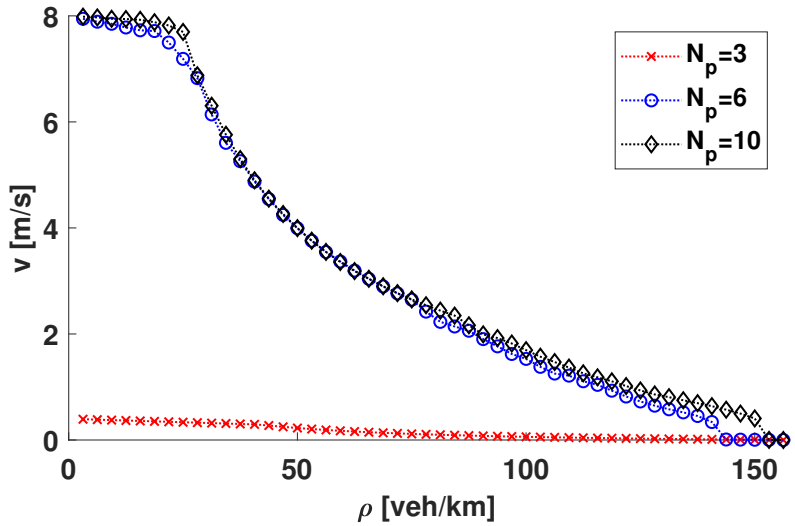


Figure 4.28: Average speeds with of baseline policy and added passing completion with $N_\mathrm{p} = \{3,\ 6,\ 10\}$ horizon lengths

before committing but to have enough free road to stop before the next obstruction after the junction. This appears as the clear stop of flows at high traffic densities in Figure 4.27 and Figure 4.28 when the policy deems unsafe the amount of free space left past the junction before the next vehicle. The tendency for this stopping phenomenon decreases with longer plans, horizon lengths, see $N_\mathrm{p} = 6$ and $N_\mathrm{p} = 10$ results. It is easy to see that, with relatively short horizon sections available to plan the crossing, vehicles are expected to have, a first high terminal speed state; the car-following constraint imposes the restricting factor, as an upper bound on the terminal speed of the follower (which is planning to cross). Concluding that the cases with longer horizons can initially cross with lower predicted terminal speeds and, thus, operate and keep continuous flows longer in high-density simulations.

In order to demonstrate how vehicle trajectories are affected, Figure 4.29 shows the

passing-completion-induced trajectories for three horizon lengths $N_p = \{6,\ 12,\ 18\}$ and two vehicles arriving at the junction. Vehicles first slow down as they arrive at the junction, until they are close enough to initiate a complete junction crossing; both vehicles may cooperate to reduce acceleration and deceleration demands during the manoeuvres.

This system works perfectly for busy junction areas where at least one other vehicle is seen on the other arm of the junction to engage with in an active junction-crossing interaction. Thus, the pairwise formulation in (4.20) has an underlying fallacy when no vehicle arrives on the other junction arm (see Figure 4.30). In the pictured case, the second vehicle did not have the passing completion constraints when it crossed the junction, because it arrived after the first vehicle completed its passing and their interaction had been deactivated and removed. Thus, even if the vehicle has a traffic jam ahead of it, it may decide to enter the junction, where it may then get stuck and block traffic from previously undetected vehicles on the other junction arm.

Most congestions form in the context of high-density traffic, which often suggests a general assumption that the other arm has a similar flow; thus, this particular fallacy case may be rare in practice. In the 8-loop simulations this particular scenario was not observed due to the junction structure. However, it may be argued that this could be beneficial if no vehicles are in the vicinity, the accompanying control actions and slow-down effect could be spared in these cases.

The fallacy of the method is the dependency on pairwise interactions, which can be alleviated if the hysteresis constraints would be generated for each agents separately. Recall the side view of the passing completion terminal set in Figure 4.26, where the side views are completely describing the allowed and restricted agent wise terminal sets. Yielding the passing-completion constraints:

$$s_p\left(k+N_\mathrm{p}|k\right)+t_\mathrm{h}v_p\left(k+N_\mathrm{p}|k\right)\leq-\frac{W}{2}+M(1-b_p) \tag{4.22a}$$

$$-s_p\left(k+N_\mathrm{p}|k\right)\leq-\frac{W}{2}-L_p+Mb_p \tag{4.22b}$$

$$\forall p\in\mathcal{N}_\mathrm{BoxAgents},$$

where the vehicles that have not left their junction conflict region are in the set $\mathcal{N}_\mathrm{BoxAgents}$, alternatively the vehicles that $s_p(k)\leq\frac{W}{2}+L_p$, from the junction geometry in Figure 4.1. Furthermore, b_p is a single binary switch implementing the passing completion like a vehicle-wise hysteresis switch. This addition to the baseline vehicle control is presented in MPC 4.6.

MPC-4.6

$$J^{*}_{\text{MPC-4.6}} = \min\ (4.1)$$

s.t. $\forall j \in \{0 \dots (N_\text{p} - 1)\},\ \forall n \in \mathcal{N}$:

$$x_n\,(k|k) = x_n\,(k)$$

$$x_n\,(k+j+1|k) = A\ x_n\,(k+j|k) + B\ u_n\,(k+j|k)$$

$$(x_n, u_n) \in \mathcal{X}_n \times \mathcal{U}_n$$

and $\forall (p, q) \in \mathcal{N}_\text{CFI}$:

(4.3)

and $\forall \{p, q\} \in \mathcal{N}_\text{JCI}$:

(4.4a)–(4.4i),

and $\forall p \in \mathcal{N}_\text{BoxAgents}$:

(4.22a)–(4.22b),

Finally, Figure 4.31 shows the cases with and without the passing-completion constraints added. In a normal case with a bigger gap in the competing traffic flow, the optimiser would take the chance and make a vehicle waiting for its right of way enter the intersection even if there is a traffic jam ahead of it. This is a valid move because, at that moment, the vehicle would not make other vehicles suffer (i.e. it would not increase the aggregated cost of the objective function but lower it because it can get closer to its destination). It is easy to see that after other vehicles appear, this move proves to be counter-productive. In contrast, when the passing-completion constraint is added, the vehicle remains in front of the junction until it can safely and completely cross; thus allowing uninterrupted traffic on the other arm.

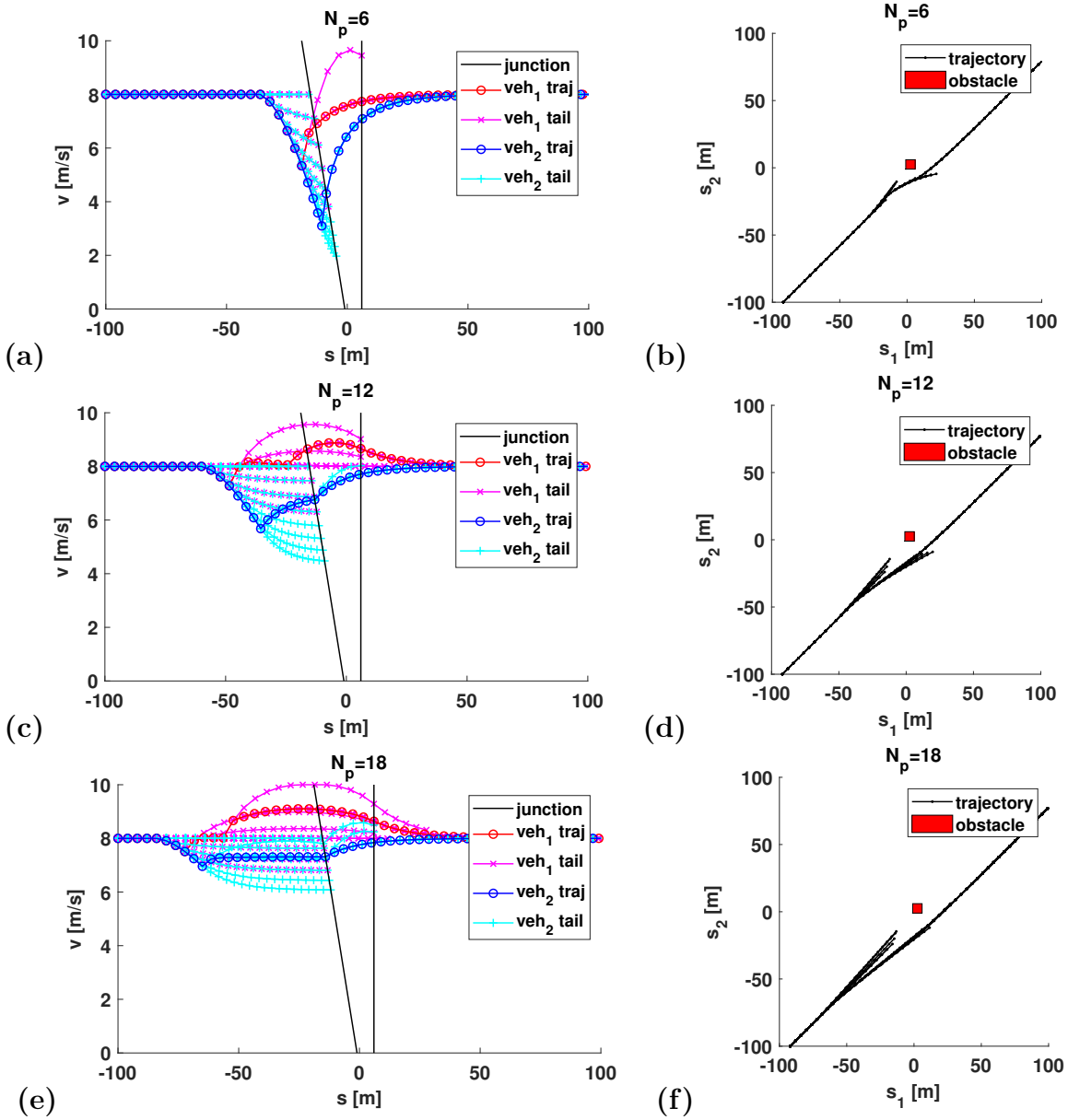


Figure 4.29: Passing completion of trajectories for $N_p = \{6,\ 12,\ 18\}$ horizon steps; the left side depicts speed–position graphs while the right side shows the related collision set graphs for two vehicles approaching the junction on different arms.

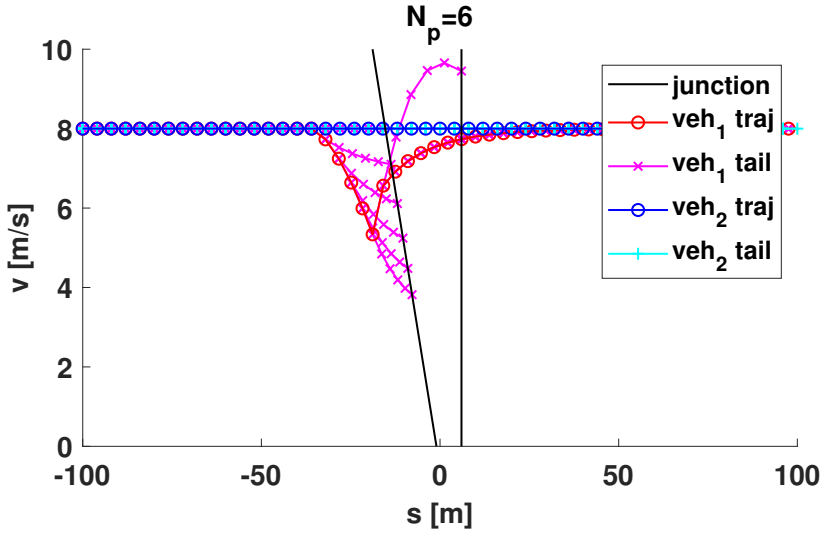


Figure 4.30: Passing completion requires an active pairwise junction-crossing interaction to generate the constraints; when the first vehicle completes its passing, the previously active junction-crossing interaction is deactivated and removed, meaning the second vehicle does not check for its passing completion.

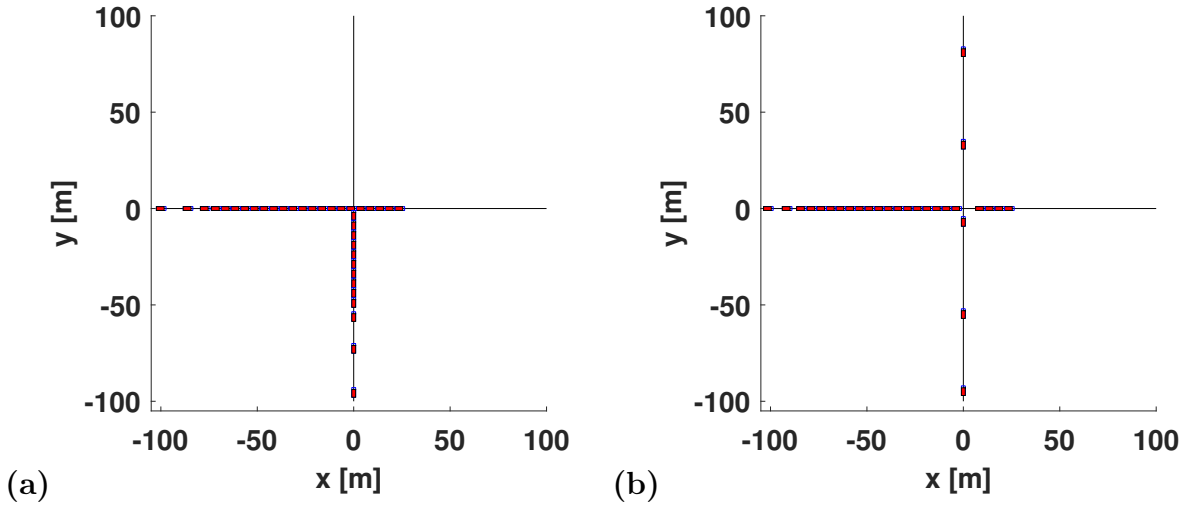


Figure 4.31: The effect of passing completion is tested by stalling a vehicle after the junction, causing a traffic jam to gradually form behind it; in **(a)**, the baseline control lets vehicles enter the junction, clogging the other junction arm as well; in **(b)**, vehicles leave the junction area clear; allowing the traffic on the other junction arm to continue uninterrupted. ($N_p = 10$)

4.7 Summary

This chapter has shown numerical simulations of multi-agent vehicle control for three policies. The baseline policy allows for junction crossing order changes and adaptation to late information. The FCFS fixed-junction-passing-order policy encodes lower computational complexity as a trade for adaptation, since it is a heuristic type approach with fewer binaries needed in the optimisation. The soft pre-avoidance policy aligns vehicles early on to pass the conflicting junction zone without heavy interaction (i.e. acceleration and deceleration) near the junction. Traffic measures were demonstrated on fundamental diagrams to compare 8-loop-junction simulations. Data shows that the chance of deadlock situations are mitigated with longer optimisation horizons, which were clearly demonstrated on a fundamental diagram with baseline control. Finally, box junction rule was implemented and demonstrated with passing-completion-terminal constraints for pair- and agent-wise formulations (i.e. do not enter the conflicting intersection region if, by the terminal state, the vehicle cannot exit it).

5

Performance and simplifications of cross junction control

The MI-MPC-based control framework, which was proposed in earlier chapters, is susceptible to reach high complexity with relatively moderate number of vehicles. This is due to the combinatoric nature of the binary decisions together with the global nature of the solver method. The intended application area is online, real-time control of autonomous vehicles. For any practical control method, it is crucial to keep the complexity and computation time limited posing bounds on the capabilities of formulations (e.g. number of vehicles considered, prediction horizon length, number of decision variables). This chapter aims to enhance the framework formulation for improving scaling and computational performance by exploiting the structure of the obstacle shape in cross junctions; exploring added constraint methods and demonstrating viability of sequential decentralisation.

5.1 Orthogonal decoupling

The collision sets on the pairwise 2D planes of conflicting vehicles can be bounded by convex polygons around the set of conflicting positions (see Figure 2.12), which extrudes to the higher-dimensional decision space (for example, see Figure 2.2). The problem feasibility is discussed in Section 2.4, which details the requirements for the angle of the obstacle bounding sides for convex deadlock-free representation. Thus, vehicles moving forward (at non-negative speeds) would be able to pass obstacle features without falling in local optima. The original obstacle polygon for cross junctions has the rectangular shape (analysed in Chapter 4) with the orientation of parallel and perpendicular sides to the position axes (see Figure 4.1).

Multiple cross-junction arbitration obstacles between crossing vehicles are represented in Figure 5.1. The vehicle with index 1 is being arbitrated against those with indices 2, 3 and 4. Note that all three collision sets share both of the bounding left-side and

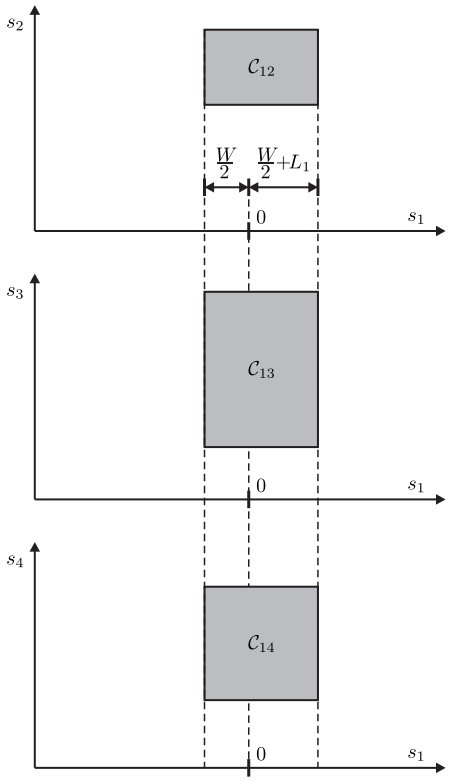


Figure 5.1: Redundant orthogonal hyperplanes between multiple collision sets.

right-side positions projected on the s_1 axis. Furthermore, when inspecting the choice of hyperplane constraints in (4.4), it is clear that they can only depend on the states of a single vehicle. Thus, if one vehicle appears in multiple junction-crossing interactions (as in Figure 5.1), repeated entries of redundant hyperplanes, which involve more new binary variables in the relaxations, are generated when using (4.4). This provides a chance for simplification.

Let a new set $\mathcal{N}_{\mathrm{JCI}_{\mathrm{agents}}}$ be generated, that collects each unique vehicle index appearing in the $\mathcal{N}_{\mathrm{JCI}}$ set of sets in order to enumerate the pairs of vehicles with active junction-crossing interactions. Thus, unique-vehicle-corresponding constraints and their big-M relaxations can now be generated as:

$$s_p\left(k+j|k\right)+t_{\mathrm{h}}v_p\left(k+j|k\right)\leq-\frac{W}{2}+Mb_{p,1}(k+j|k) \tag{5.1a}$$

$$-s_p\left(k+j|k\right)\leq-\frac{W}{2}-L_p+Mb_{p,2}(k+j|k) \tag{5.1b}$$

$$s_p\left(k+j+1|k\right)+t_{\mathrm{h}}v_p\left(k+j+1|k\right)\leq-\frac{W}{2}+Mb_{p,1}(k+j|k) \tag{5.1c}$$

$$-s_p\left(k+j+1|k\right)\leq-\frac{W}{2}-L_p+Mb_{p,2}(k+j|k) \tag{5.1d}$$

$$\forall p\in\mathcal{N}_{\mathrm{JCI}_{\mathrm{agents}}},\quad \forall j\in\{0\dots(N_{\mathrm{p}}-1)\}.$$

The constraint of separating hyperplane theorem is still defined on the $\mathcal{N}_{\mathrm{JCI}}$ (active

junction-crossing interactions) to connect vehicle pairs:

$$b_{p,1}(k+j|k) + b_{p,2}(k+j|k) + b_{q,1}(k+j|k) + b_{q,2}(k+j|k) \leq 3 \qquad (5.2)$$
$$\forall \{p,q\} \in \mathcal{N}_{\mathrm{JCI}}, \quad \forall j \in \{0 \dots (N_{\mathrm{p}}-1)\}.$$

In summary, the controller gets the following form:

MPC-5.1

$J^*_{\text{MPC-5.1}} = \min$ (4.1)
s.t. $\forall j \in \{0 \dots (N_{\mathrm{p}}-1)\},\ \forall n \in \mathcal{N}$:

$$x_n(k|k) = x_n(k)$$
$$x_n(k+j+1|k) = A\, x_n(k+j|k) + B\, u_n(k+j|k)$$
$$(x_n, u_n) \in \mathcal{X}_n \times \mathcal{U}_n$$

$\forall (p,q) \in \mathcal{N}_{\mathrm{CFI}}$:

(4.3)

$\forall p \in \mathcal{N}_{\mathrm{JCI_{agents}}}$:

(5.1a)–(5.1d)

and $\forall \{p,q\} \in \mathcal{N}_{\mathrm{JCI}}$:

(5.2)

and $\forall p \in \mathcal{N}_{\mathrm{BoxAgents}}$:

(4.22a)–(4.22b).

5.2 Time-window allocation

Treating junctions as resource-allocation problems is an attractive choice that has been extensively explored by researchers. An intersection-occupancy time-slot-scheduling problem is shown in [34], with additional economic objectives employing primal decomposition. This technique was later used in [35] to obtain the approximate optimal time slots, iteratively considering the vehicle dynamics. In this section, the previously used formalism is applied to create a time-window-allocation approach for the problem over its finite horizon. The numerous binary inequalities of separating hyperplane theorems ((5.2) $\forall \{p,q\} \in \mathcal{N}_{\mathrm{JCI}}$) may be condensed into one separating hyperplane relation through the use of the orthogonal decoupling property of the cross junction.

Assume that vehicle-wise decoupled inequalities are in place from (5.1). There are two cases. In the first case, the vehicle is either in front of or past its junction conflict

region. This case concludes that one of the two hyperplanes must be active (non-relaxed), in terms of binaries:

$$b_{p,1}(k+j|k) + b_{p,2}(k+j|k) = 1. \tag{5.4}$$

In the second case, suppose that if a vehicle is allowed to occupy the shared road section within the junction, it would need to relax both of its safety hyperplanes:

$$b_{p,1}(k+j|k) + b_{p,2}(k+j|k) = 2. \tag{5.5}$$

Now, consider that only one vehicle can safely have the right of way in the junction at any one time; thus, all other vehicles must remain outside of the junction conflict region, yielding:

$$\sum_{p \in \mathcal{N}_{JCI\text{agents}}} b_{p,1}(k+j|k) + b_{p,2}(k+j|k) = (N_{JCI\text{agents}} - 1) + 2$$

or, alternatively expressed with an inequality:

$$\sum_{p \in \mathcal{N}_{JCI\text{agents}}} b_{p,1}(k+j|k) + b_{p,2}(k+j|k) \leq (N_{JCI\text{agents}} - 1) + 2, \tag{5.6}$$

where the number of vehicles is $N_{JCI\text{agents}}$ in the set of $\mathcal{N}_{JCI\text{agents}}$.

This separating hyperplane theorem implements a junction-crossing sequencing as though it were resource allocation of a shared medium (junction). However, it is more than a simple resource allocation; this formulation can readily incorporate vehicle dynamics, time-headway safety, operating limits and any other constraints that preserve the problem structure. Moreover, in this decision-making, the consideration of vehicle cooperation is directly incorporated through the optimised cost of the aggregated objective function. Figure 5.2 shows the allocated discrete time windows; each time window can only occupy a given time span alone. The structure of the corner-cutting prevention inflates the size of the time windows to prevent junction entry between the discrete time steps while the constraints are only defined at the discrete time steps.

The final forms of the time-window constraints for the MPC are summarised in MPC 5.2.

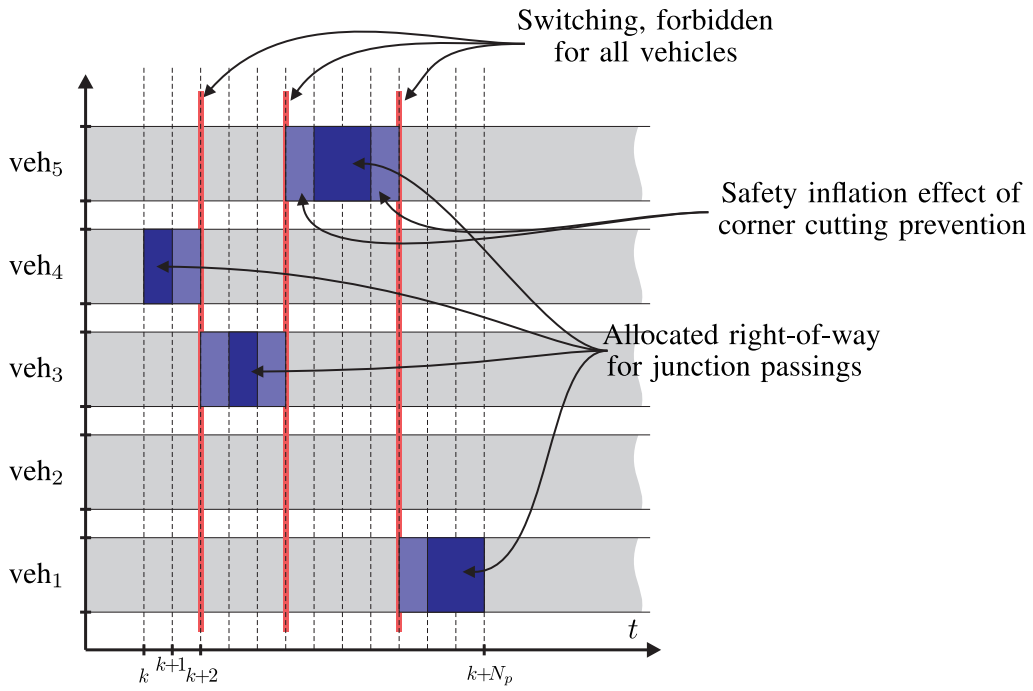


Figure 5.2: Time-window allocation schematic where the vehicles have the right of way granted over the prediction horizon; there is a moment of discrete time separation between different allocation windows; the inflation around the time window is the effect of the corner-cutting prevention formulation, which does not allow for double entry into the junction over continuous time.

MPC-5.2

$$J^*_{\text{MPC-5.2}} = \min\ (4.1)$$

s.t. $\quad \forall j \in \{0 \dots (N_\text{p} - 1)\},\ \forall n \in \mathcal{N}$:

$$x_n\,(k|k) = x_n\,(k)$$

$$x_n\,(k+j+1|k) = A\ x_n\,(k+j|k) + B\ u_n\,(k+j|k)$$

$$(x_n, u_n) \in \mathcal{X}_n \times \mathcal{U}_n$$

$\forall (p,q) \in \mathcal{N}_{\text{CFI}}$:

(4.3)

$\forall p \in \mathcal{N}_{\text{JCI}_{\text{agents}}}$:

(5.1a)–(5.1d)

and $\forall p \in \mathcal{N}_{\text{BoxAgents}}$:

(4.22a)–(4.22b)

and :

(5.6).

5.3 Numerical tests of formulations

In this section, three numerical 8-loop simulations will be compared—MPC 4.5, MPC 5.1 and MPC 5.2. MPC 4.5 is the un-simplified (US) baseline vehicle-control formulation with passing completion hysteresis; MPC 5.1 is the formulation of the orthogonal decoupling (OD) from Section 5.1; MPC 5.2 is the time-window (TW) allocation from Section 5.2. The characteristic length d_c was chosen to attain the desired traffic densities $\rho = \{50,\ 100,\ 120\}\ \frac{\text{veh}}{\text{km}}$ for $N = 10$ vehicles in the 8-loop junction. The simulations were deterministic; they started from the same initial states and ran for all three vehicular controls with each new parameter value. The simulated time was chosen to be 10 minutes (= 600 s) with $\delta t = 0.5$ s. Moreover, the deadlock-averting or box-junction-implementing terminal junction-passing-completion constraints were added for all formulations. The un-simplified formulation is the baseline problem with (4.4) junction constraints and (4.20) box-junction passing completion. The orthogonal decoupling is a simplified formulation composed of the (5.1) agent-wise junction constraints coupled with the (5.2) binary separating-hyperplane constraints of and the (4.22) agent-wise box-junction implementation. The time-window allocation is based on the (5.1) agent-wise junction constraints coupled with the (5.6) binary separating-hyperplane inequality and the (4.22) agent-wise box-junction implementation.

A distance measure was used to analyse whether decisions are the same or different across formulations. Simulation similarity was evaluated through a comparison of vehicle trajectories between run pairs. The comparison was done through a cumulated squared position error measure:

$$\epsilon = \sum_{n \in \mathcal{N}} \left[\sum_{k=0}^{k_{\max}} (s_n^{\text{sim a}}(k) - s_n^{\text{sim b}}(k))^2 \right]^{\frac{1}{2}}, \tag{5.8}$$

where the maximum number of discrete simulation time steps is $k_{\max}$ and the simulation results sim a and sim b are compared and stored for post-processing. Alternatively, differences between trajectories may be identified by looking at the average flow Q measures, which depend on the distances travelled by vehicles and are presented in Table 5.1, Table 5.2 and Table 5.3. A dissimilar trajectory simulation is shown in Figure 5.3, which presents a single selected trajectory across the two simulations, where Q average flow was also different.

It was determined that trajectories with a short horizon control and low traffic density ($N_p = 6$ and $\rho = 50\ \frac{\text{veh}}{\text{km}}$) yield a low number of interactions (trajectory changes due to obstacle avoidance) in the simulations; the trajectories were identical for all three methods in this region (US, OD and TW). With the same horizon length, however, only the US and OD formulations remained close to one another (i.e. $\rho = 100\ \frac{\text{veh}}{\text{km}}$: $\epsilon \approx 0.0081$ and $\rho =$

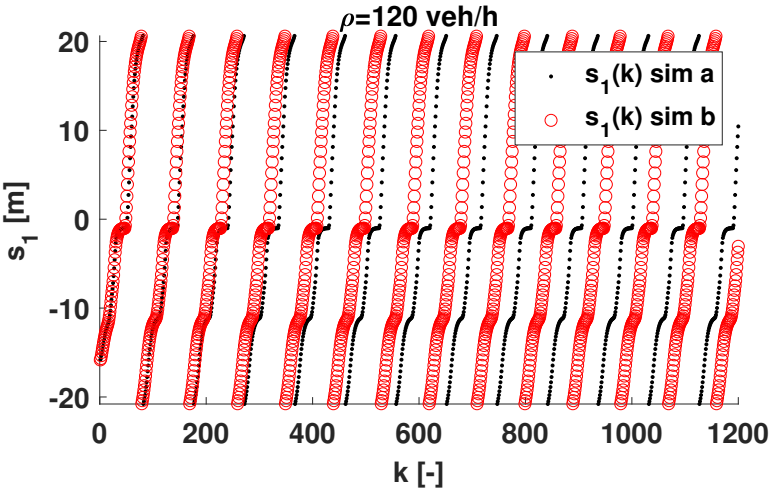


Figure 5.3: Vehicle trajectories from Table 5.3 of $N_p = 6$ horizon black dots present the vehicle positions for un-simplified formulation while the red circles are presenting the time-window formulation achieving higher average flow in in the 8-loop

120 $\frac{\text{veh}}{\text{km}}$: $\epsilon \approx 0$). The TW resulted different trajectories ($\epsilon \neq 0$) for $\rho = \{100,\ 120\}$ $\frac{\text{veh}}{\text{km}}$.

In contrast to the intuition that these simplifications should yield identical results, in fact, no evidence was found indicating that these formulations give identical solutions in general. Dissimilar results dominated for the time-window allocation formulation even though car-following interactions and the junction safety constraints with corner-cutting prevention were shown to have been satisfied when checked in post-processing. The US and OD simulations provided similar results and performance but towards longer horizon lengths they diverged away.

The integration of noise, inherent in numerical methods and algorithms, leads to the build-up of absolute error over long simulations. In every simulation time step small but dissimilar integrated noise was suspected to be the reason that US and OD simulations reached different states while their inner logic should have yielded the same answers. In order to test if the solutions remained the same in terms of single control steps in time, the following method was implemented. The outputs of a single controller were used in the simulation loop, though other controllers parallel to it also did control calculations. All controllers have been fed from the same input source in each control time step and all of their output results have been inspected. This procedure was expected to eliminate the slight drift mismatch from the noise integration between the simulations to verify that the control formulations are yielding practically identical results and that the solver process noise is non-dominant in a single time step scale. It became clear that the diverging phenomenon (bifurcation) starts at the end of the prediction tails and begins to diverge slightly on the next control inputs to be issued. As time progresses, the states gradually bifurcate and diverge, opening a difference gap between trajectories that is no longer negligible.

Origins of this diverging phenomenon could not be clearly identified due to the com-

plexity of the problem. The trivial cause could be the symmetric nature of the problem—for example, recall the case in which the vehicles are distributed evenly and symmetrically in the initial configuration with zero speed (Figure 4.21 **(a)**). At this point, slight noise in the solution process could yield the same patterns but develop in a mirrored way on the symmetric loops. Even the slightest noise could trigger the other solution to be chosen, such as the multiple definitions of the constraints in the US formulation compared to the OD without them. To avoid this possibility, a unique minimal position shift was added to each initial vehicle configuration to prevent a symmetric case. Of course, this does not remove the existing decision-splitting effect in the problem (i.e. initially discussed for the merging decisions, see Figure 2.15 and Figure 2.17 decision graphs).

Furthermore, sensitivity can also play a role, as it is generally grows higher for the longer horizons. Consider this analogy of threading a needle with the thread as the future prediction tail and the needle as an obstacle providing multiple discrete decision options for avoidance. The further the thread is held from its end, the more difficult it is to correctly thread it through the hole of the needle (corresponding to a given discrete choice previously taken by the other formulation).

The same reason may stand behind the non-identical results of the controller formulations, where the high sensitivity of the decisions at the end of the horizon tail practically incorporates slight ambiguity in the problem. Furthermore, pushing this thread (receding the horizon tail forward in time) enlarges the slight position ambiguity of diverging tries (which are the opposite sides of the bifurcation) and enforces the outcome even more; similarly the forking propagates through the horizon states for the vehicles.

Type ($\rho = 50$ $\left[\frac{\text{veh}}{\text{km}}\right]$)	Horizon	Computation time t_{opt} [ms]				Q $\left[\frac{\text{veh}}{\text{h}}\right]$
		Mean	Max.	Min.	Std.	
US	$N_p = 6$	15.99	29.10	6.34	3.84	719.69
	$N_p = 12$	229.12	406.34	89.02	43.67	719.49
	$N_p = 18$	3099.38	6427.48	1300.95	714.58	718.52
OD	$N_p = 6$	14.09	25.47	5.46	3.75	719.69
	$N_p = 12$	125.67	201.85	51.66	23.01	718.52
	$N_p = 18$	650.05	1132.60	376.31	148.37	718.52
TW	$N_p = 6$	13.82	30.36	5.38	3.77	719.69
	$N_p = 12$	125.49	196.16	53.69	21.88	719.49
	$N_p = 18$	541.67	1083.63	336.86	129.30	718.51

Table 5.1: Computational time and flows are shown for 600 s simulations; US, OD and TW are tags for, respectively, un-simplified, orthogonally decoupled and time-window formulations with $N_p = \{6,\ 12,\ 18\}$ horizons in the 8-loop; number of cars was 10 and the $\rho = 50$ $\frac{\text{veh}}{\text{km}}$

Across the three tables (Table 5.1, Table 5.2 and Table 5.3), it is clear that the un-simplified problem takes the longest to solve. It is also evident that computational cost increases with the horizon length. On closer inspection, the trend is exponential; this is in line with the expected exponential growth of combinations. The number of binaries is

Type ($\rho = 100$ $[\frac{\text{veh}}{\text{km}}]$)	Horizon	Computation time t_{opt} [ms]				Q $[\frac{\text{veh}}{\text{h}}]$
		Mean	Max.	Min.	Std.	
US	$N_p = 6$	16.50	38.43	9.98	5.90	552.57
	$N_p = 12$	228.60	443.73	101.37	66.28	598.77
	$N_p = 18$	3113.00	7793.13	996.78	1258.91	624.57
OD	$N_p = 6$	12.96	41.79	8.34	5.90	552.57
	$N_p = 12$	138.19	250.64	74.78	27.69	598.77
	$N_p = 18$	607.65	1246.44	342.45	143.59	624.57
TW	$N_p = 6$	13.13	35.06	7.82	4.80	553.14
	$N_p = 12$	139.76	245.70	74.87	31.00	608.78
	$N_p = 18$	574.75	1018.68	354.82	102.10	624.57

Table 5.2: Computational time and flows are shown for 600 s simulations; US, OD and TW are tags for, respectively, un-simplified, orthogonally decoupled and time-window formulations with $N_p = \{6,\ 12,\ 18\}$ horizons in the 8-loop; number of cars was 10 and the $\rho = 100$ $\frac{\text{veh}}{\text{km}}$

Type ($\rho = 120$ $[\frac{\text{veh}}{\text{km}}]$)	Horizon	Computation time t_{opt} [ms]				Q $[\frac{\text{veh}}{\text{h}}]$
		Mean	Max.	Min.	Std.	
US	$N_p = 6$	16.62	44.48	8.73	6.75	378.76
	$N_p = 12$	183.15	338.77	78.27	54.58	463.78
	$N_p = 18$	1592.57	4691.45	558.31	654.56	490.16
OD	$N_p = 6$	13.51	45.71	7.86	5.84	378.76
	$N_p = 12$	124.69	230.54	62.22	32.10	463.40
	$N_p = 18$	494.65	962.97	271.73	107.96	490.79
TW	$N_p = 6$	17.11	43.77	7.79	7.56	399.30
	$N_p = 12$	130.09	246.37	53.60	34.49	494.04
	$N_p = 18$	517.49	1172.62	318.59	111.18	494.23

Table 5.3: Computational time and flows are shown for 600 s simulations; US, OD and TW are tags for, respectively, un-simplified, orthogonally decoupled and time-window formulations with $N_p = \{6,\ 12,\ 18\}$ horizons in the 8-loop; number of cars was 10 and the $\rho = 120$ $\frac{\text{veh}}{\text{km}}$

in a linear relation with the length of horizon.

The OD and TW approaches yield consistently higher computational speeds across all test cases. [29] also found that when axes decouple in the MPC formulation, faster solving times can be observed. Computational times are shown for the OD and TW formulations with quantile box representation in Figure 5.4, Figure 5.5 and Figure 5.6. The figures supplement the tables through additional statistical information of computational times, such as the median, $25-75\%$ quantile box, and whiskers. The outlier computation times are those that the whiskers cannot envelop, indicating the long tail of the distribution. These outliers are either attributed to rare, computationally difficult cases or are simply considered to be artefacts of delays in task scheduling common in the consumer operating system.

The TW approach sees slight differences across the different traffic densities. The OD approach has better computational performance at higher traffic densities than does the TW approach.

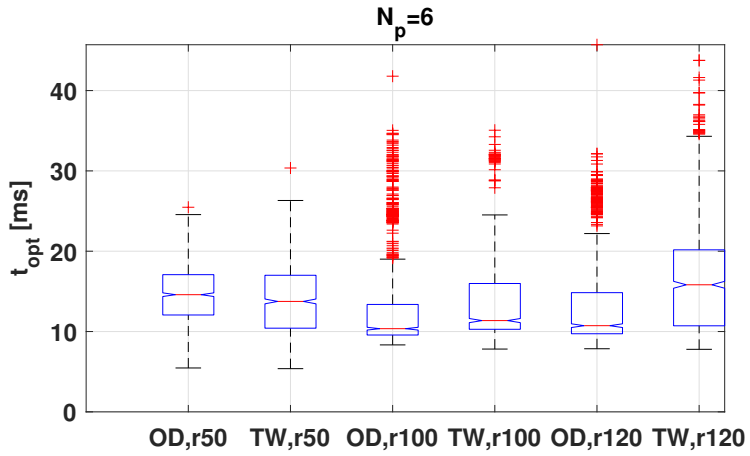


Figure 5.4: Computation time distributions of $\rho = \{50,\ 100,\ 120\}\ \frac{\text{veh}}{\text{km}}$ simulations for $N_p = 6$ horizons in the 8-loop junction; OD and TW are tags for orthogonally decoupled and time-window approaches, respectively, numbers after tags refer to the ρ traffic density; distributions are visualised with $25-75\%$ quantile boxes and with whiskers that are, at maximum, 1.5 of the box length; the red '+' marks are outliers outside the extent of the whiskers.

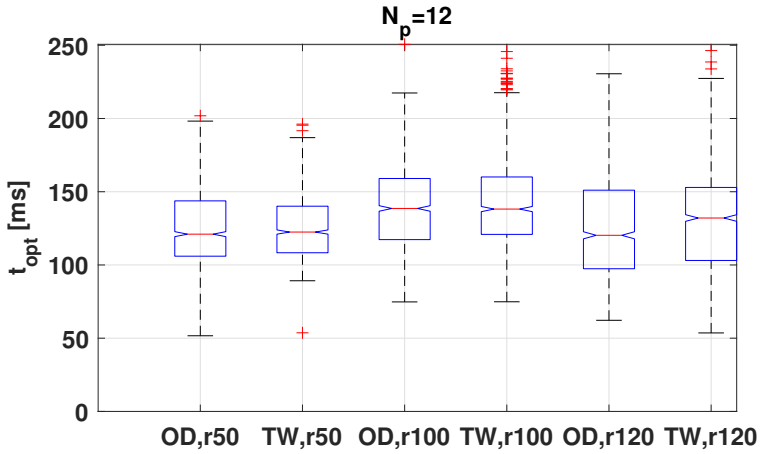


Figure 5.5: Computation time distributions of $\rho = \{50,\ 100,\ 120\}\ \frac{\text{veh}}{\text{km}}$ simulations for $N_p = 12$ horizons in the 8-loop junction; OD and TW are tags for orthogonally decoupled and time-window approaches, respectively, numbers after tags refer to the ρ traffic density; distributions are visualised with $25-75\%$ quantile boxes and with whiskers that are, at maximum, 1.5 of the box length; the red '+' marks are outliers outside the extent of the whiskers.

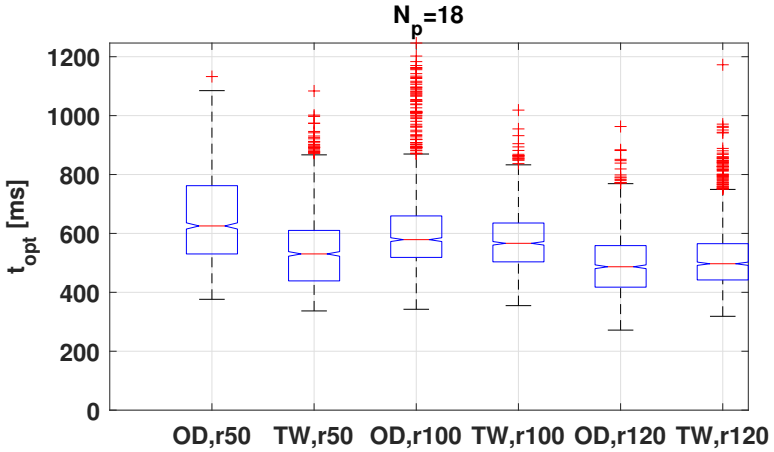


Figure 5.6: Computation time distributions of $\rho = \{50,\ 100,\ 120\}\ \frac{\text{veh}}{\text{km}}$ simulations for $N_p = 18$ horizons in the 8-loop junction; OD and TW are tags for orthogonally decoupled and time-window approaches, respectively, numbers after tags refer to the ρ traffic density; distributions are visualised with $25-75\%$ quantile boxes and with whiskers that are, at maximum, 1.5 of the box length; the red '+' marks are outliers outside the extent of the whiskers.

5.4 Improving efficiency with added binary constraints

Generally, the more binaries applied, the more combination are to be considered while searching for the global optimum (in this case, the number of convex sub-problems to be treated). Namely, where the number of binaries is n_b, 2^{n_b} cases are to be explored with brute-force approaches. This may be a major influencing factor in the computational speed of the hybrid-MPC (MIQP) optimisations presented in this work. However, according to [82], it is the tightness of the formulation that puts a practical limit on the number of integer configurations that are to be explored. More sophisticated solver approaches may be able to exploit this rule (e.g. with branch-and-bound). By adding certain extra constraints, usually based on a relation not evident to the solver, some help is provided to the solver that may discover short cuts and simplifications in an easier and faster manner; two types of approaches that are explored here.

First, adding redundant binary constraints is aimed to exclude sub-problems in another, more direct way by providing an alternate and simpler path to the solver. In this way, the full feasible domain of the original formulation coincides with the one of the new formulation regardless the new extra redundant constraint being added. This may spare time by reaching earlier conclusion or feasibility check for a binary combination without going through the entire sub-problem evaluation.

Second, another approach is by simply adding constraints which may not be redundant, and thus, partially remove some previously valid feasible sub-domains (sub-problems). The possible drawback is the chance of excluding the global optimum of the original problem or even casting it infeasible. This way the number of sub-problems to be evaluated becomes fewer with the fewer feasible binary combinations but may trade or compromise solution quality.

In the following extra inequalities will be added to the OD problem MPC 5.1 and TW problem MPC 5.2. These constraints will be cast on the binaries and aimed to exploit the strong causality between the steps in the finite time horizon, junction passing related logic and their binary relaxation. Later the car-following interactions will be used to add redundant constraints on binaries. With the aim to take into account junction crossing arbitration and car-following when multiple vehicles do arbitration from the same lane.

5.4.1 Added binary causality constraints

Suppose that a vehicle only allowed one transition in temporal sense between relaxed and non-relaxed constraints of different modes (i.e. intersection approaching, and intersection left).

Such added logic would remove some degree of freedom from the system, limiting the options within scenarios but reducing the computational complexity in the combinatorial sense as well as promising computational performance increase.

	k	$k+1\|k$	$k+2\|k$	$k+3\|k$	$k+4\|k$	$\cdots$	$k+N_\mathrm{p}-1\|k$	$k+N_\mathrm{p}\|k$
b_app	0	0	1	1	1	1	1	1
b_left	1	1	1	1	0	0	0	0

Table 5.4: Schematic of the evolution of binary relaxations for-junction approaching and junction-left constraints (0: active; 1: relaxed) over the finite horizon of a vehicle; grey cells correspond to the junction passing where both constraint types are relaxed.

A simple case of the evolution of junction approaching and junction left constraint relaxation, and related binaries are demonstrated in Table 5.4. The shown case is kept simple to remain practical and easy to visualise the process of transitions. For this reason, the simplification applies only when t_h is chosen close to zero, removing velocity dependency in (4.4) or (5.1) junction passing constraints. This would cast these constraints to 2D purely position based ones and, furthermore, uses $s(k+j+1|k) \geq s(k+j|k)$ relation from the non-negative speed model. On the example case (see Table 5.4) a distinct monotonic increasing (decreasing) rule can be observed for the junction approaching (junction left) hyperplane relaxation. This monotonic rule is coupled to the monotonic increasing and progressing nature of vehicle position in time.

Thus, the monotonic rule, coded as inequalities, yields in the original cross junction binary formulation:

$$b_{pq,1}(k+j|k) \leq b_{pq,1}(k+j+1|k) \tag{5.9a}$$
$$b_{pq,2}(k+j|k) \geq b_{pq,2}(k+j+1|k) \tag{5.9b}$$
$$b_{pq,3}(k+j|k) \leq b_{pq,3}(k+j+1|k) \tag{5.9c}$$
$$b_{pq,4}(k+j|k) \geq b_{pq,4}(k+j+1|k) \tag{5.9d}$$
$$\forall\{p,q\} \in \mathcal{N}_\mathrm{JCI}, \quad \forall j \in \{0 \dots (N_\mathrm{p}-2)\}$$

or for the agent wise decoupled formulation:

$$b_{p,1}(k+j|k) \leq b_{p,1}(k+j+1|k) \tag{5.10a}$$
$$b_{p,2}(k+j|k) \geq b_{p,2}(k+j+1|k) \tag{5.10b}$$
$$\forall p \in \mathcal{N}_{JCI\mathrm{agents}}, \quad \forall j \in \{0 \dots (N_\mathrm{p}-2)\},$$

with $N_\mathrm{p}-1$ new tightening binary inequalities per original relaxations.

Now to show that some degree of freedom is lost by this added constraint formulation (based on $t_\mathrm{h} \cong 0$ assumption) recall that the second order vehicle dynamics may allow higher rate of deceleration than the conservative linear position-velocity, safe simple time-headway, based hyperplane would allow. This means that a vehicle may enter into a safe passing phase to later return back into safe slow down phase and give the right of way to a different vehicle in the collision set pair. This concept is shown in Figure 5.7 with junction approaching binaries taking a non-monotonic transition in time. Figure 5.8 shows that

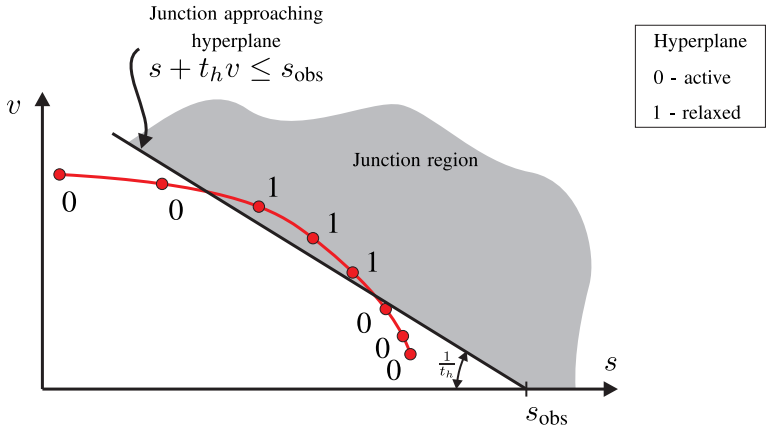


Figure 5.7: Added binary constraint states over the finite horizon plan of a vehicle that may enter the junction region (subject to the safety of other vehicles) but, due to its dynamics, it can return to its own safe junction-approaching region before the junction; this means that while a vehicle may take advantage of the right of way for junction-crossing it has no immediate plan to go ahead and complete the crossing. (s_{obs} indicates the beginning of the junction conflict region at zero speed).

the new formulation (e.g. MPC 5.3) only allows the monotonic transition of relaxations. This means that if a vehicle has been assigned the right of way it cannot cancel it later on in the plan even if the vehicle dynamics would yield some benefits to this action; loosing this degree of freedom effectively reducing the number of binary configurations. MPC 5.3 shows how these added binary constraints fit into the previous control framework on the example of the orthogonal decoupling from Section 5.1.

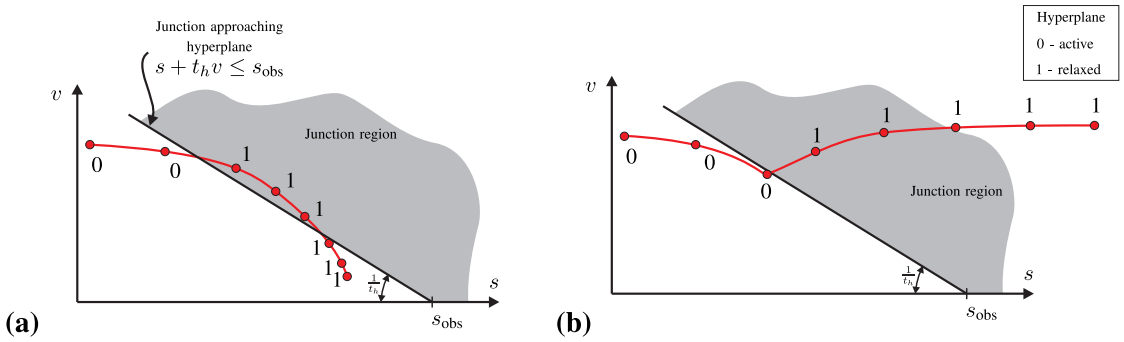


Figure 5.8: Finite horizon plans with indicated binary states where, due to the constraints added, after the vehicle has relaxed its safe junction-approaching hyperplane, it may enter the junction region (subject the safety of other vehicles) or, as shown in (a), return the set of states previously satisfying the junction approaching-hyperplane. However, it remains the sole holder of the right of way within the rest of the current plan, and it may only be given away once the vehicle leaves the junction region. In contrast, without the added constraints, the vehicle would be able to cancel its right of way (see Figure 5.7). (b) shows a healthy crossing where the tightening may make no apparent change (s_{obs} indicates the beginning of the junction conflict region at zero speed).

MPC-5.3

$$J^*_{\text{MPC-5.3}} = \min \ (4.1)$$

s.t. $\forall j \in \{0 \dots (N_{\text{p}} - 1)\}, \ \forall n \in \mathcal{N}$:

$$x_n\,(k|k) = x_n\,(k)$$

$$x_n\,(k+j+1|k) = A\ x_n\,(k+j|k) + B\ u_n\,(k+j|k)$$

$$(x_n, u_n) \in \mathcal{X}_n \times \mathcal{U}_n$$

$\forall (p,q) \in \mathcal{N}_{\text{CFI}}$:

(4.3)

$\forall p \in \mathcal{N}_{\text{JCI}_{\text{agents}}}$:

(5.1a)–(5.1d)

and $\forall \{p,q\} \in \mathcal{N}_{\text{JCI}}$:

(5.2)

and $\forall p \in \mathcal{N}_{\text{BoxAgents}}$:

(4.22a)–(4.22b)

and $\forall p \in \mathcal{N}_{\text{JCI}_{\text{agents}}}, \quad \forall j \in \{0 \dots (N_{\text{p}} - 2)\}$:

(5.10a)–(5.10b).

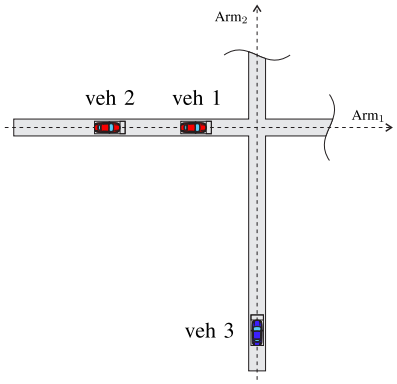


Figure 5.9: Scenario for added binary constraints, which may incorporate a car-following relation; two vehicles engaged in a car-following interaction must arbitrate with a third vehicle on the other junction arm to cross the junction.

5.4.2 Added car-following-related binary constraints

Suppose that multiple vehicles approach a junction on one arm while multiple other vehicles are approaching the junction on the other arm. Due to the car-following constraint, a follower vehicle cannot cross the junction before its leader vehicle. This rule may provide an opportunity to add extra relations purely on binaries, similar to the previously discussed monotonic rule (Section 5.4.1). This time, however, the binary inequalities are defined between follower and leader vehicles and no degrees of freedom are expected to be lost, since this junction traffic property is a substantial requirement, already fulfilled by the car-following interactions.

Figure 5.9 shows an example situation for which the junction-approaching binaries of the two vehicles are indicated in Figure 5.10. By noticing the inter-vehicle relaxation rule between the vehicles, the following inequalities can be formulated:

$$b_{p,1}(k+j|k) \leq b_{q,1}(k+j+1|k) \tag{5.12a}$$

$$b_{p,2}(k+j|k) \geq b_{q,2}(k+j+1|k) \tag{5.12b}$$

$$\forall (p,q) \in \mathcal{N}_{\text{CFI}},\ p \in \mathcal{N}_{JCI\text{agents}},\ q \in \mathcal{N}_{JCI\text{agents}},$$
$$s_q(t) \geq s_p(t), \quad \forall j \in \{0 \dots (N_\text{p} - 2)\},$$

where the follower vehicle is p and the leader vehicle is q; both are using the agent-wise formulation related to the binary relaxations of junction-approaching and junction-left constraints in (5.1). MPC 5.4 shows how these added binary constraints fit into the previous control framework on the example of the orthogonal decoupling from Section 5.1.

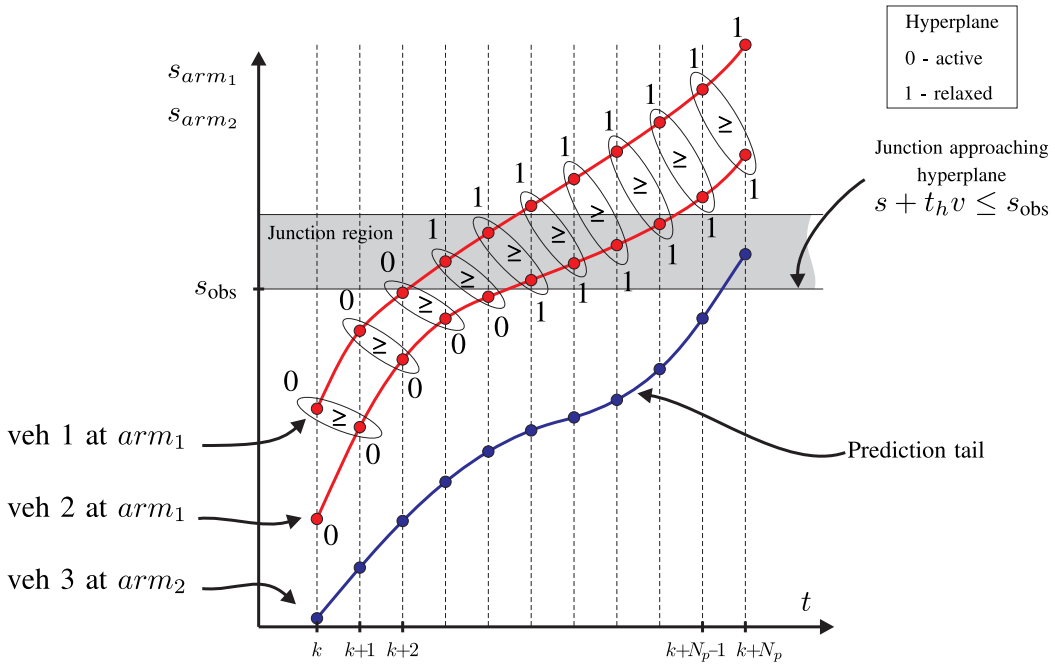


Figure 5.10: Two vehicles engaged in a car-following interaction must arbitrate with a third vehicle on the other junction arm to cross the junction (see situation in Figure 5.9). The collision set binaries of vehicles 1 and 2 have added binary constraint formulation by defining binary inequalities in accordance with the car-following interaction. This case only pictures the junction-approaching binaries and their relations; due to the spatio-temporal car-following, the constraints also have a time-step shift in their relations (only for this 2D plane illustration $t_\text{h} \approxeq 0$ to make the junction beginning representable by a line).

MPC-5.4

$$J^*_{\text{MPC-5.4}} = \min \ (4.1)$$

s.t. $\quad \forall j \in \{0 \dots (N_\text{p} - 1)\}, \ \forall n \in \mathcal{N}$:

$$x_n\,(k|k) = x_n\,(k)$$

$$x_n\,(k+j+1|k) = A\ x_n\,(k+j|k) + B\ u_n\,(k+J|k)$$

$$(x_n, u_n) \in \mathcal{X}_n \times \mathcal{U}_n$$

$\forall (p,q) \in \mathcal{N}_{\text{CFI}}$:

(4.3)

$\forall p \in \mathcal{N}_{\text{JCI}_{\text{agents}}}$:

(5.1a)–(5.1d)

and $\forall \{p,q\} \in \mathcal{N}_{\text{JCI}}$:

(5.2)

and $\forall p \in \mathcal{N}_{\text{BoxAgents}}$:

(4.22a)–(4.22b)

and $\forall (p,q) \in \mathcal{N}_{\text{CFI}},\ p \in \mathcal{N}_{\text{BoxAgents}},\ q \in \mathcal{N}_{\text{BoxAgents}},\ \forall j \in \{0 \dots (N_\text{p} - 2)\}$:

(5.12a)–(5.12b).

5.5 Numerical tests with added binary constraints

The added binary constraint cases are shown only for the OD and TW formulations, as they have better computational performances than the US formulation, as shown in Section 5.3. Furthermore, only the horizon lengths of $N_p = 6$ and $N_p = 12$ are considered because the data in Section 5.3 revealed that $N_p = 18$ would yield considerably longer computational times than the control time step $\delta t = 0.5$ s for $N = 10$ vehicles, which would not be appropriate for real-time control in this case. The test cases are the base problem formulations without the added binary constraints, the two single cases shown in Section 5.4.1 and Section 5.4.2, and the test cases in which both types of these binary constraints are added.

Appendix A presents the computational time and average flow results collected in Table A.1, Table A.2 and Table A.3. The results indicate that, in most cases, the singe-agent related added binary constraints from Section 5.4.1 resulted in lower flows due to the removed degree of freedom. There was a slight improvement in a few cases but none that was consistent for all computation times. The car-following related added binary constraint method shown in Section 5.4.2 did not result in any significantly different flows than the cases without it; however, it increased the overhead and calculation time because the increased number of constraints in the optimisation. In fact, the added binary constraints of this kind are already implied by the car-following constraints indirectly but the solver (Gurobi) has been able to consider this indirect link in its branch-and-bound algorithm, solving the mixed-integer program. This means that, since the complexity remained the same but computation time and problem size increased, there were only negative effects for this approach.

A similar idea using car-following orders to enhance the control performance was mentioned in [1]; however, in contrast to the theorised computational benefits there, the data suggests that the implementation analysed here yields increased computational time and problem size.

The last case is that which employs both added binary constraint formulations together, which inherits both the lower flow measure from the reduced degree of freedom and the overhead time from the car-following-related added binary constraints. This case consistently reached solution times in the range of computation times measured for the cases that only applied one of the added binary constraint techniques.

5.6 Decentralisation

In this section, the predictive control will be formulated for smaller but connected sub-problems, exchanging calculated plans between agents in sequential order. Sequential decentralisation was discussed for a predictive multi-agent system in [62]. Centralised approaches scale poorly and hard to tackle above a certain size; decentralisation is an attractive way to fix scaling issues of a problem especially for large systems [49]. Scattolini [69] surveys the distributed MPC formulations and notes that vehicle-related problems are usually coupled through obstacle avoidance constraints and the sub-systems are decoupled in dynamics. Dunbar and Murray [23] show an early work for stabilisation of multiple-vehicle formation, where sub-systems are coupled through the cost. In [76], the agents are coupled and interacting via constraints in a tube-MPC approach. This accommodates agents with added uncertainty to safely travel in their tubes and uses sequential replanning, through which agents update one plan at a time. In the case of road vehicle control, Debada et al. [21] demonstrate a cooperative framework for sharing different levels of information between vehicles, including future MPC plans to improve the traffic throughput in junctions and roundabouts.

Shi et al. [71] illustrate a decentralised solution framework to conform with MPC like behaviour, ensuring a rear-end-collision-free system while new vehicles on arrival are added to the problem.

The rear-end-collision-free property in the decentralised formulation shown here is handled by the simple time-headway safety constraints and the upstream sequencing when hard prediction tails are shared.

5.6.1 Problem formulation

In this problem, the vehicles are only able to gain information about their surroundings from within a given vicinity. For example, assume that a vehicle in question (ego vehicle) is moving into a cross junction (see Figure 5.11). The detection area within which the ego vehicle is able to gather information is a square-shaped area centred on the front of the vehicle box extending in all direction with the length of range l_R (see Figure 5.11). Note that this is the detection area where the range on the 2D plane is calculated in an l_∞-norm sense (the range in an l_2-norm sense would be a circular area). From the viewpoint of the ego vehicle n, vehicles in this detection area are elements of the in-range set $\mathcal{N}_{\text{inR}}(n)$.

$$\mathcal{N}_{\text{inR}}(n) = \left\{ p \in \mathcal{N} \mid \left| \begin{bmatrix} x_p(k) \\ y_p(k) \end{bmatrix} - \begin{bmatrix} x_n(k) \\ y_n(k) \end{bmatrix} \right|_{l_1} \leq l_R \right\}, \tag{5.14}$$

where the 2D Cartesian coordinates are calculated from the map (2.2) or, alternatively, from the 1D position in (2.3).

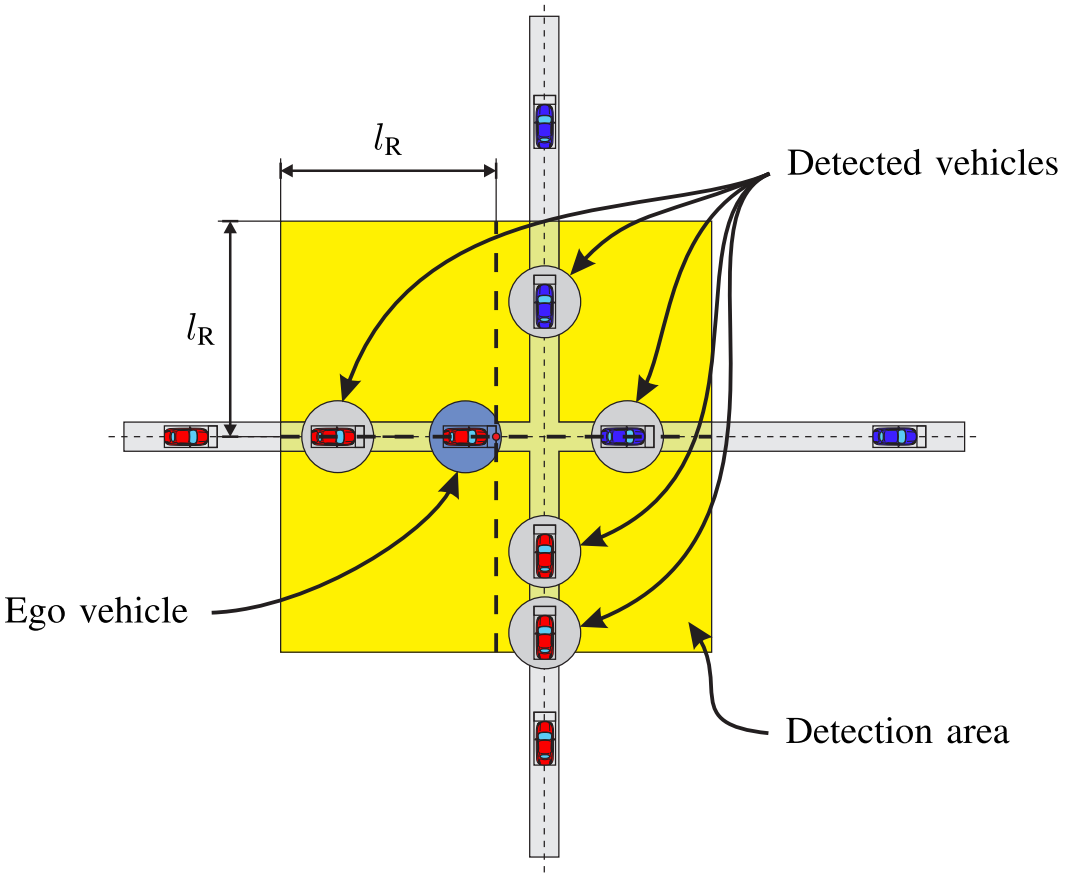


Figure 5.11: Detection area of an ego vehicle, with a square-shaped detection area.

The sequence in which the vehicles optimise is obtained from their order of appearance at the arm inlets. Vehicles follow the control sequence chosen upstream, opposite to the direction of traffic, (vehicles closest to the outlet end of the road optimise first, and vice versa). This is important because vehicles have only non-negative speeds; hypothetically, if the plans were calculated the other way around, the front vehicle could be forced into an unwanted or unsafe situation (similar to tail gating).

Let the sequence be $\mathcal{S}$; the i-th element of this partially ordered set is the individual car $\mathcal{S}_i \in \mathcal{N}$, $i = \{1 \dots N\}$. This defines the order in which the optimisations are solved and provides a hierarchical structure to the control problem. The appearance time of the vehicles is used to determine the sequence; for vehicle n this is $t_{\text{app } n}$. The appearance time is a known because it is pre-generated and sampled from the inlet distribution (see Section 4.3.3 and Appendix B). Alternatively, it can be determined from the vehicle positions:

$$t_{\text{app } n} = \min\{t \mid s_n(t) = s_{\text{entry}}\}, \tag{5.15}$$

where the junction arm entry position is s_{entry}. Note that the vehicles are entering through the same inlet position (s_{entry}) but on different junction arms. The sequence starts with the index of the vehicle, which appears the earliest; each next element is defined in (5.16),

Algorithm 1 Vehicle control

Obtain vehicle measurements $x_n(k)$, $\forall n \in \mathcal{N}$
Initialise sequence counter $i = 1$
for $n \in \mathcal{S}_i$ **do**
  Receive broadcasted solutions in range, $x_p^*,\ u_p^*$, $\forall p \in (\mathcal{N}_{\text{inR}}(n) \cap \mathcal{N}_{\text{Tx}})$
  Solve optimisation (e.g. MPC 5.5) with vehicles $\mathcal{N}_{\text{inR}}(n)$
  Transmit solution x_n^*, u_n^* in effective radius
  Increment sequence counter $i = i + 1$
end for
return $u_n^*(k|k)$, $\forall n \in \mathcal{N}$ to simulation engine

as follows:

$$\mathcal{S}_1 = \arg\min_n \ t_{\text{app } n}, \tag{5.16a}$$

$$t_{\text{app } n=\mathcal{S}_i} \leq t_{\text{app } n=\mathcal{S}_{i+1}}, \qquad i \in \{1 \dots (N-1)\}, \tag{5.16b}$$

which makes $\mathcal{S}$ a partially ordered set that orders vehicles in an upstream sequence on their respective road.

The vehicles that already optimised (in the current time step) broadcast without delay their own trajectory choices over their finite horizon $x_p^*,\ u_p^*$, $p \in \mathcal{N}_{\text{Tx}}(k)$, where the set of vehicle indices that already transmitted their results at k control period is $\mathcal{N}_{\text{Tx}}(k)$. The next vehicle to optimise with index n is able to receive broadcast results of other vehicles and fix these prediction tails in its own optimisation if the transmissions were in range, $\forall p \in (\mathcal{N}_{\text{inR}}(n) \cap \mathcal{N}_{\text{Tx}})$. When both prediction tails of a pair of vehicles to be considered in an optimisation are fixed, the interaction constraints between them are removed from this optimisation because they are already satisfied.

Vehicle interactions within range of and with respect to the individual ego vehicle n are formulated with the help of the set 'vehicles in range' $\mathcal{N}_{\text{inR}}(n)$, namely $\mathcal{N}_{\text{CFI}}(n)$, $\mathcal{N}_{\text{JCI}}(n)$, $\mathcal{N}_{\text{JCI}_{\text{agents}}}(n)$ and $\mathcal{N}_{\text{BoxAgents}}(n)$; these sets available for the ego vehicle, for simplicity, are obtained from the global sets in the simulation by considering the in-range condition; for example:

$$\mathcal{N}_{\text{CFI}}(n) = \{(p, q) \in \mathcal{N}_{\text{CFI}} \mid p \in \mathcal{N}_{\text{inR}}(n),\ q \in \mathcal{N}_{\text{inR}}(n)\}. \tag{5.17}$$

Following the above logic, $\mathcal{N}_{\text{JCI}}(n)$ can be generated while $\mathcal{N}_{\text{JCI}_{\text{agents}}}(n)$ represents the set of unique vehicle indices inside $\mathcal{N}_{\text{JCI}}(n)$.

The structure of the MPC is detailed for the decentralisation framework in MPC 5.5 for ego vehicle n, the original formulation was based on the orthogonal decoupling example.

Finally, Algorithm 1 shows the vehicle control at k control time step with the sequential optimisation.

MPC-5.5

$$J^*_{\text{MPC-5.5}} = \min \sum_{p \in \mathcal{N}_{\text{inR}}(n)} w_p \left(\sum_{j=0}^{N_\text{p}-1} \left[q\xi_p(k+j|k)^2 + r u_p(k+j|k)^2 \right] + q_\text{f} \xi_p(k+N_\text{p}|k)^2 \right)$$

s.t. $\forall j \in \{0 \dots (N_\text{p}-1)\},\ \forall p \in \mathcal{N}_{\text{inR}}(n),\ p \notin \mathcal{N}_{\text{Tx}}$:

$$x_p(k|k) = x_p(k)$$
$$x_p(k+j+1|k) = A\, x_p(k+j|k) + B\, u_p(k+j|k)$$
$$(x_p, u_p) \in \mathcal{X}_p \times \mathcal{U}_p$$

$\forall p \in \mathcal{N}_{\text{inR}}(n),\ p \in \mathcal{N}_{\text{Tx}}$:

$$x_p(k|k) = x_p(k)$$
$$x_p(k+j+1|k) = x^*_p(k+j+1|k)$$
$$u_p(k+j|k) = u^*_p(k+j|k)$$

$\forall (p,q) \in \mathcal{N}_{\text{CFI}}(n),\ p \notin \mathcal{N}_{\text{Tx}},\ q \notin \mathcal{N}_{\text{Tx}}$:

(4.3)

$\forall p \in \mathcal{N}_{\text{JCI}_\text{agents}}(n),\ \exists \{p,q\} \in \mathcal{N}_{\text{JCI}}(n),\ p \notin \mathcal{N}_{\text{Tx}},\ q \notin \mathcal{N}_{\text{Tx}}$:

(5.1a)–(5.1d)

and $\forall \{p,q\} \in \mathcal{N}_{\text{JCI}}(n),\ p \notin \mathcal{N}_{\text{Tx}},\ q \notin \mathcal{N}_{\text{Tx}}$:

(5.2)

and $\forall p \in \mathcal{N}_{\text{BoxAgents}}(n),\ p \notin \mathcal{N}_{\text{Tx}}$:

(4.22a)–(4.22b).

5.6.2 Numerical tests

In the simulations, a centralised control case for baseline results and two types of decentralised approaches are considered. The first decentralised approach is the one described in Section 5.6.1, in which the prediction horizons are shared between vehicles within range. The second decentralised approach is when the prediction horizons are only shared when the vehicles are within the vicinity of the junction (within the detection area; see Figure 5.12). In this case, $\mathcal{N}_{\text{Tx}}$ is redefined with an added position requirement:

$$\mathcal{N}_{\text{Tx}} = \left\{ p \in \mathcal{N}_{\text{Tx}} \mid \left\| \begin{bmatrix} x_p(k) \\ y_p(k) \end{bmatrix} - \begin{bmatrix} x_\text{junc} \\ y_\text{junc} \end{bmatrix} \right\|_{l_1} \leq l_R \right\}, \tag{5.19}$$

where the coordinates of the junction are x_junc and y_junc. In this case, vehicles are cooperating through messages close to the junction, where the organised action is crucial

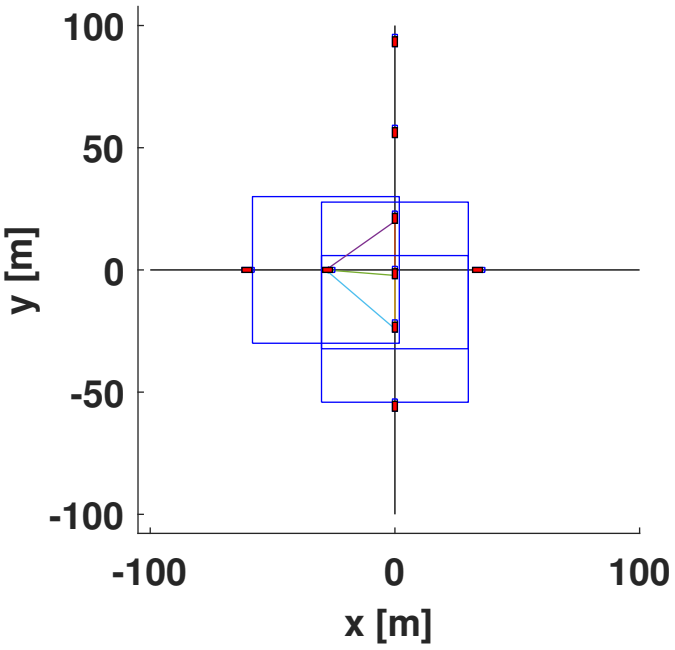


Figure 5.12: Prediction horizon tails shared between vehicles shown with connecting lines; ego vehicles, which received shared results are indicated by their square detection areas.

for the right-of-way arbitration. This could alleviate the burden of sequencing for many vehicles in the junction area—the more vehicles there are in the sequence, the longer it would take to complete all individual optimisations due to sequential dependencies. For example, the vehicles far from the junction (farther than l_R) are not required to wait for communication; they may do their optimisation parallel without cross-dependency. This decouples vehicle decisions outside of the junction vicinity.

The previously shown OD and TW approaches were tested for the decentralised control. For OD, the modifications to obtain the decentralised MPC formulation were presented in MPC 5.5; for TW, the same steps have been taken, and the formulation include only vehicles and interactions within the detection area from the viewpoint of the ego vehicle. TW formulation has exhibited sensitive behaviours, especially for circular detection ranges with the decentralised formulation, likely due to the partial view of the cross junction, which occasionally led to infeasibility. The detection area was chosen to be square-shaped, to ease the sensitivity of the decentralised TW, which provided better results. Even with this choice, however, the decentralised TW reached an infeasible agent optimisation at high traffic densities after a long simulation length $t \approx 35$ minutes for both TW decentralisation approaches. Considering the rarity of this event it is difficult to profile its origin.

In Figure 5.13, simulation results are shown as colour-coded densities instead of being depicted through the immense number of vehicle trajectories. The simulations were run for a 3600 s session for OD and TW formulations with $N_p = 12$ horizon length, $l_R = 30$ m and added passing completion constraints. The symmetric inlets were injecting vehicle flows with each distribution defined with the parameters of minimum 10 $\frac{\text{veh}}{\text{h}}$, maximum

1400 $\frac{\text{veh}}{\text{h}}$ and mean 500 $\frac{\text{veh}}{\text{h}}$, corresponding to medium average flows with low traffic density. Desired speeds were randomly chosen with uniform distribution from the $[7,\ 9]$ ms^{-1} range.

In Figure 5.13 a small but clear speed dip can be seen on the left side, this originates from the passing completion condition, which makes the vehicles, arriving to the junction, initiate a slow-down phase because, at this point, it is impossible to plan the junction passing since the vehicle cannot yet verify if it can leave the junction. There are no significant differences between the OD and TW formulations. The centralised solution achieves the best system-wide cost while the two decentralised approaches show increased control action closer in the junction. Consequently, after it is verified that the junction crossing can be completed, the vehicles speed up in the decentralised formulation, because the detection area of the ego vehicles do not yet cover the junction. When a vehicle gets close to the junction, it can start to arbitrate with other vehicles in range resulting in a second slow-down region for the decentralised formulations. This second slow-down phase yields lower speeds before the junction than in the centralised case. A slight difference can be seen between the cases where all possible horizons are shared in an upstream direction and those where they are restricted to the vicinity of the junction. This yields slightly lower minimum average speeds before the junction, which is in line with the missed opportunity to obtain more downstream information.

Figure 5.14 displays congested traffic cases where both inlet distributions were set to parameters corresponding to high traffic density—minimum 10 $\frac{\text{veh}}{\text{h}}$, maximum 1400 $\frac{\text{veh}}{\text{h}}$ and mean 1300 $\frac{\text{veh}}{\text{h}}$. Both decentralised TW formulations reached infeasibility at the same simulation point that being $t \approx 35$ minutes. This is attributed to the sensitivity of TW rather than vehicles colliding; the conflicting hard constraints in the problem render this control infeasible. The unexpected issue was showing conflicts in mid-horizon, leading to the conclusion that either the sequence order was inappropriate for the formulation or the numerical rounding errors aggregated on the shared plans until a hard constraint was violated. This means that the orthogonal decoupling, which shows better robustness properties, is favoured for use as a decentralised approach.

This congested case meant that the inlet could not achieve the desired vehicle injection flow because there was occasionally no available free space in which to place the new vehicles. However, the junction had developed a stable and continuous but low throughput. The number of vehicles that could not be introduced at the inlets due to the congestion were discarded but registered. As an analysis method, the registered number of discarded vehicles were summed up road-wise (on junction arm_1 and arm_2) for a final comparison (see Table 5.5). The close numbers within the pairs show that the roads were having a symmetric average flow with similar inlet distributions. Evidently, both roads, on average, arbitrated in a fair manner.

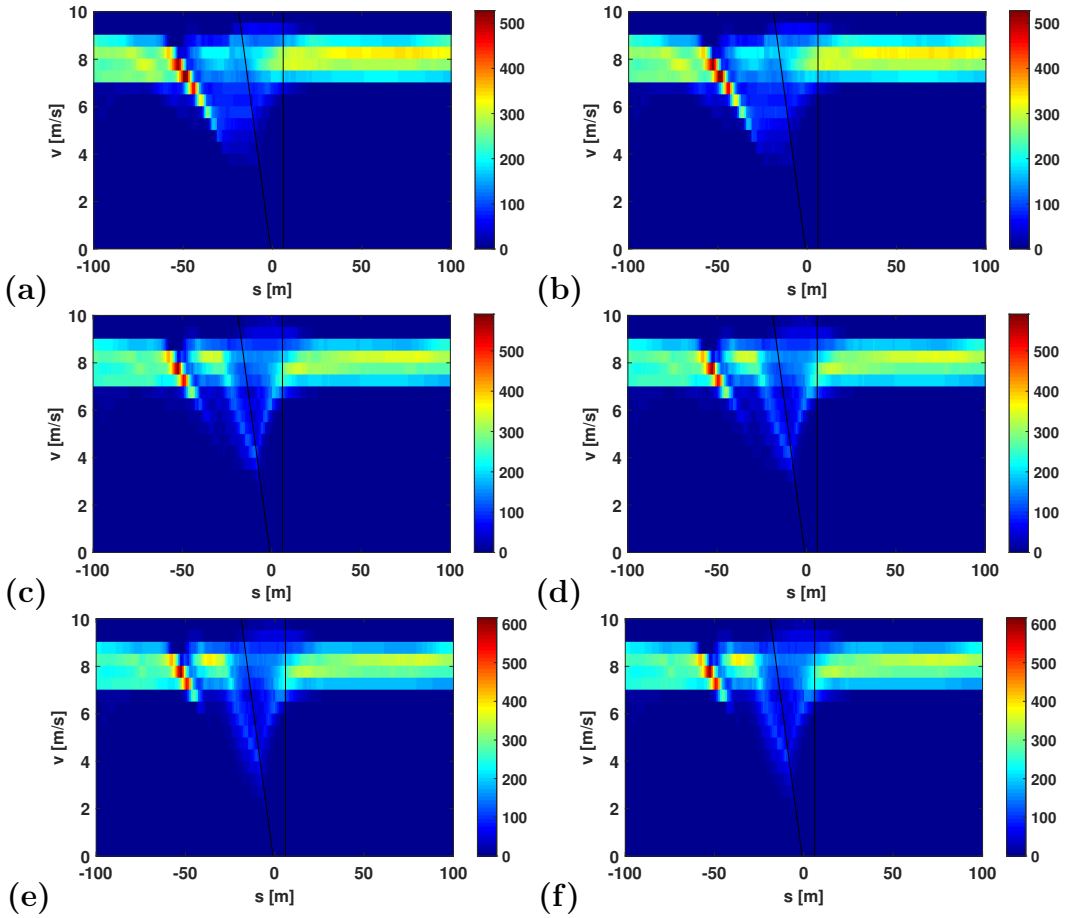


Figure 5.13: Vehicle trajectory densities for OD in **(a)**, **(c)** and **(e)** and for TW formulation in **(b)**, **(d)** and **(f)**; **(a)** and **(b)** show the centralised results; **(c)** and **(d)** show the decentralised results with prediction-horizon sharing; **(e)** and **(f)** show the the decentralised results with horizon sharing only at the junction.

	centralised	decentralised 1	decentralised 2
OD	576 – 577	574 – 574	570 – 574
TW	576 – 565	332 – 333	333 – 331

Table 5.5: The pairs of numbers correspond to the numbers of discarded vehicles on junction arm_1 and junction arm_2 throughout the each simulation. Distributed TW simulations were stopped at 35 minutes; for all other cases the simulation times correspond to 60 minutes of traffic. The even numbers within the pairs show, on average, fair junction arbitration.

Furthermore, in the aggregated trajectory density graphs shown in Figure 5.14, the vehicles travelled with a stationary speed in the congested road section before they could arbitrate their junction-crossing orders. Recall that the speed-position figures are generated as overlaid trajectories from both junction arms; since the flow of vehicles as the trajectory densities are coinciding and no disjoint density peaks can be seen on the figures, this allows the conclusion that the junction arm flows are fair, thus, the arbitration is fair. For the centralised control, it is easy to see the existence of cooperation and fairness; if more vehicles are on a junction arm than the other and they are moving slowly then their cost of not travelling at their desired speed is high and aggregated. This makes the more

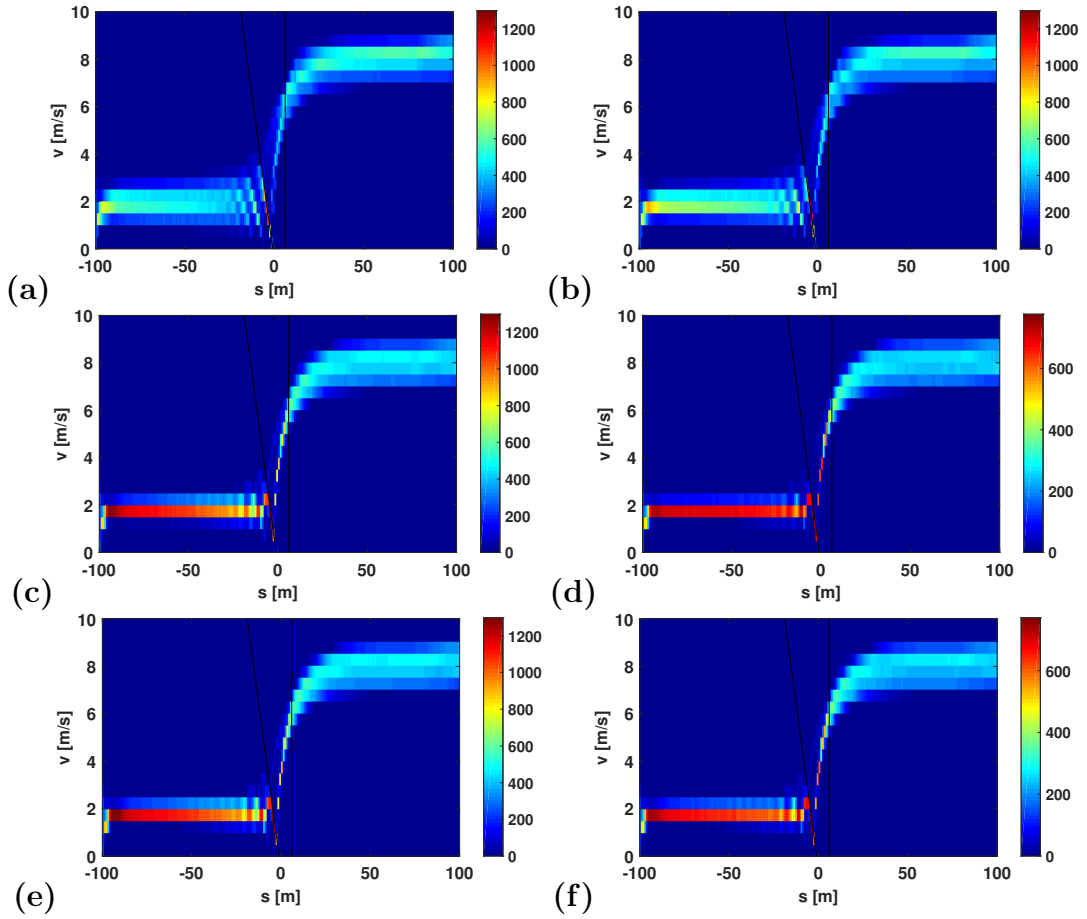


Figure 5.14: Congested vehicle trajectory densities for OD in **(a)**, **(c)** and **(e)** and for TW formulation in **(b)**, **(d)** and **(f)**; **(a)** and **(b)** show the centralised results; **(c)** and **(d)** show the decentralised results with prediction horizon sharing; **(e)** and **(f)** show the decentralised results with horizon sharing only at the junction; in **(d)** and **(f)** the results are only until the distributed TW formulations encountered an infeasible optimisation at $t \approx 35$ minutes.

fortunate higher-flow traffic be considerate of those on the other junction arm and grant the congested arm the right of way.

Imagine the situation of ego vehicles travelling on congested junction arms in the decentralised framework. The detection area of vehicles spans all around them even behind of them (recall the shape of the detection area, Figure 5.12). It may be argued that there is no specific reason to include vehicles behind the ego vehicle in the optimisation; however, this allows a cost pressure to be generated on the vehicles. As with the centralised case when the flow on one junction arm suffers more than that on the other, the arbitration follows cooperative fairness proven also by the balanced drop numbers in Table 5.5. This fairness element makes the method more like a locally centralised control scheme while still remaining decentralised overall.

5.7 Summary

This chapter discussed performance enhancements and improved scaling formulations based on agent-wise hyperplanes exploiting the structure of the cross junction obstacle shapes in the collision sets. The removal of redundant hyperplanes and their binary variables allowed for the achievement of higher optimisation speeds. With using the same considerations, a time-window allocation approach was shown to operate with just a single connecting separating-hyperplane binary relation. Two added binary causality constraint approaches were shown, in which reducing the degree of freedom led to slightly lower computation times at the cost of lower flows in dense traffic. Furthermore, by adding binary constraints, that incorporate car-following relations into junction arbitration revealed no benefit and only increased the overhead time. Finally, the decentralised formulation proved the emergence of cooperative fairness in the cross junction arbitration and traffic flows with sequentially shared MPC predictions between vehicles. In this approach, orthogonal decoupled decentralisation was found to cope well in the simulations.

6
Concluding remarks

In Chapter 2, a mixed-integer MPC relaxation formulation was developed for centralised autonomous driving with positive control-invariant safety constraints based on the time-headway parameter for Y-junction merging. Corner-cutting prevention was added to account for the worst-case leader-vehicle dynamics, making no optimistic assumptions about leader dynamics for one control time step. The recursive feasibility of the MPC algorithm was theoretically, and later numerically, verified. Decision graphs were created showing the dependency of decisions on inter-vehicle gaps and relative priority weighting. These vehicle-order decisions were shown to be dependent on the distance from the merging point and, after a limit distance, they settled to a periodic relation with underlying phase property. In line with expectations, numerical examples with multiple vehicles showed that changing relative vehicle weightings affected the merging order, and the control framework accomplished vehicle cooperation.

Chapter 3 inspected the cost function of the control method, where the stability proof could be shown for an l_1-norm terminal-cost-based objective function. Furthermore, the l_1-norm-based formulation has a soft behaviour that can be tuned to coincide with the 'running-cost' formulation with hard terminal set constraints of simple time-headway safety, also providing the stability proof for the later formulation. This is due to the 'running-cost' being one of the three cases from the decomposed l_1-norm form. Furthermore, the tuning and choice of the simple time headway give the vehicle trajectories a distinct near-obstacle deceleration that was found to safeguard obstacle avoidance for very short horizon controls and give a softer slow-down-approach phase for longer horizons. To promote earlier merging actions than what the horizon length would dictate and to smoothly comb vehicles from the merging junction arms, a soft pre-merging heuristic was formulated to heuristically minimise future vehicle-state conflicts through added cost terms. Additionally, minimal and maximum speed bound regions were formulated with the safe time headway and corner-cutting prevention using the mixed-integer formulation.

Cross-junction multi-agent simulations were tackled in Chapter 4, which largely ad-

dressed the numerical considerations for and aspects of the problems. Three policies were shown: baseline, FCFS and soft pre-avoidance. The baseline policy has the advantage of full adaptability in vehicle-order changes while, in contrast, FCFS used pre-fixed vehicle-order heuristics, allowing lower computational cost but less adaptability. The soft pre-avoidance policy showed increased control actions before getting near the junction and less intense vehicle interactions near the junction region with, on average, higher possible crossing speeds achievable through the junction. To avoid potential junction-blocking configurations, which could result in a deadlock situation, box-junction related constraints were formulated to prevent vehicles from starting their crossing manoeuvre if they are unable to complete it within their finite horizon plan.

Chapter 5 first inspected a performance increase stemming from the removal of multiple definitions of redundant obstacle constraints. This is because the structure of the cross-junction problem allows for simplifications in the general case with orthogonal decoupling between vehicles. Moreover, using the same hyperplanes, a time-window resource allocation policy was formulated with regard to the previous considerations for the problem structure. Two redundant binary constraint techniques were added. One removed some degrees-of-freedom using problem causality and, thus, slightly lowered computational cost (but in some cases also decreased the traffic flow). The other used car-following considerations in junction-crossing interactions; this failed to give any practical performance increase, adding only overhead to the problem. Finally, decentralised control approaches were shown, in which every vehicle has a restricted field of perception and range of communication. Vehicles were able to maintain safe junction crossings with the orthogonal-decoupling formulation but failed to do so in a rare event employing time-window formulation. The key element is that, before committing to cross the junction, vehicles must agree on the right-of-way by sharing sequentially their own and considering received predicted plans.

6.1 Future works

Future works may use the proposed mixed-integer MPC framework for high-quality predictive vehicle control even in real autonomous driving cases by extending the formulation with robust control tools to guard against uncertainties and cast the problem robustly feasible. An application of the control to real-world cases may require additional elements of efficient relaxations [38] and highly tailored MPC algorithms [81] that exploit problem structure, warm start and early termination. Furthermore, good heuristics can prime the branch-and-bound search, increasing computation speed through techniques such as shared (networked) perception and deep learning.

The framework introduced in this thesis is concerned with the atomic junction blocks. In real-world structures, several of these atomic blocks (e.g. mergings, crossings, and

multi-lanes roads) are present, which could be handled by an enhanced version of the proposed control algorithm. Furthermore, it is noted that the predictive nature of the control resembles to that of human driving; however, the exact parameters of objective function and the length, accuracy and resolution of the horizon in human processing and decision-making are potential research topics. In that case, using predictive control models to simulate traffic in road networks could closely capture the real traffic tendencies of conventional, non-autonomous vehicles.

In more detailed future works, specific features could and should be improved to obtain better efficiency.

In Y-junction merging, for example, constraint redundancy could be exploited with the orthogonal decoupling between vehicles from Chapter 5. This could increase computational performance and allow for a greater number of vehicles to be considered in real time. Using the decentralised framework could further increase the scaling possibilities of this method.

Additional techniques from mixed-integer programming (e.g. move blocking) and further binary tightening could be used to find trends and trade-offs between prediction quality and control-performance quality. Warm starting and early terminating the method could also be beneficial by creating a custom-made solver algorithm with further features added, such as penalising changes to previously predicted plan states.

Within the decentralised framework, the vehicles sharing their plans and acting differently in the junction only caused problems on a few occasions. This could indicate that, not all intentions need to be continuously communicated in cases where these intentions are clear and decided by natural opportunities in the traffic. Thus, only a very limited amount of communication may be necessary for deciding vehicle-crossing orders even in intense traffic. In human-driving scenarios, the limited amount of communication usually takes the form of meta-communication of intentions with a light beam, carefully waiting, and trying (slow-rolling) arbitration or driver head and gaze tracking. Employing an objective to maintain space and time clearance from situations requiring arbitration and communication has the chance to form similar result to the soft pre-avoidance policy, which partially achieved a distinct clearance from conflicting states all-together, resulting in more fluid junction crossings.

As an extension of the box junction, a better junction-approaching phase can be implemented with anticipating dynamically changing visibility (field-of-view) through the horizon and tailoring the passing completion feature in the decentralised case; an out-of-range vehicle cannot be detected through either radio communication or visual observation, but speculatively expected to be detectable further ahead in the prediction. The implementation could follow the idea of shadow-region method in [56]. Early results showed that in cluttered junction areas, (see Figure 6.1), flow might suffer due to vision-blocking elements

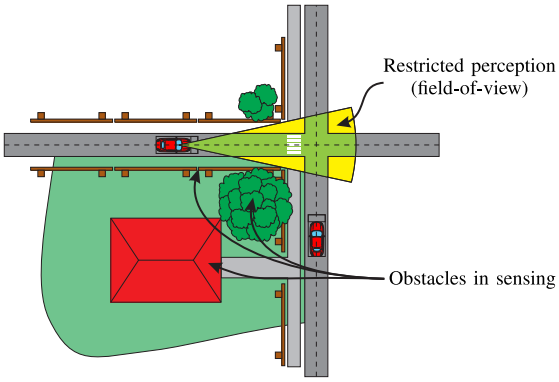


Figure 6.1: Cluttered junction area that requires a careful approach due to a reduced field-of-vision.

in the environment that force drivers to assume worst-case scenarios about the traffic on the intersecting junction arm in each time step. In summary a confident driving style could be chosen when a speculated optimistic traffic situation is assumed further ahead the horizon close to the intersection and where the gradual increase of the field-of-view is considered as the function of predicted junction proximity. The dynamic change in the field-of-view over the prediction horizon can also affect the plans in wider applications, for example, if the communication link strength is incorporated in the control problem where the millimetre-wave communication bears propagation properties resembling to the properties of visible light. It is expected for this reason that the environment in junction surroundings, affecting the field-of-view, has direct impact on junction bandwidth in general.

Furthermore, high-fidelity vehicle communication models could be imposed on top of the MPC vehicle control, adding varying delays on the information exchange. Improvement on model details could be formulated considering link strength and bandwidth, acquiring a link earlier to exchange plans and horizons, exchanging far-ahead traffic states, predicting and evaluating future traffic affecting route choices.

Future works, in summary, are promising in the field of automation and the proposed mixed-integer MPC framework could mark the beginning and act as the basis of a new tier of autonomous vehicle researches.

A

Appendix: Data tables

A.1 Added binary constraints results

Type ($\rho = 50$ $[\frac{\text{veh}}{\text{km}}]$)	Binary constraints	Horizon	Computation time t_{opt} [ms]				Q $[\frac{\text{veh}}{\text{h}}]$
			Mean	Max.	Min.	Std.	
OD	-	$N_p = 6$	14.51	106.68	5.56	5.39	719.69
	a	$N_p = 6$	13.03	42.61	5.52	5.15	719.69
	b	$N_p = 6$	14.28	26.63	5.48	3.93	719.69
	ab	$N_p = 6$	13.48	34.03	5.59	5.66	719.69
TW	-	$N_p = 6$	13.96	26.05	5.57	3.99	719.69
	a	$N_p = 6$	12.87	34.22	5.61	5.28	719.69
	b	$N_p = 6$	13.85	26.24	5.40	3.88	719.69
	ab	$N_p = 6$	13.22	33.16	5.55	5.16	719.69
OD	-	$N_p = 12$	132.10	231.18	57.92	25.28	719.49
	a	$N_p = 12$	132.77	258.43	71.72	34.72	719.49
	b	$N_p = 12$	143.36	249.74	66.98	27.19	719.49
	ab	$N_p = 12$	137.30	259.50	73.95	33.92	719.49
TW	-	$N_p = 12$	132.47	270.91	53.84	24.15	719.49
	a	$N_p = 12$	139.65	244.18	72.61	40.80	719.49
	b	$N_p = 12$	129.31	207.65	71.63	20.69	719.49
	ab	$N_p = 12$	134.77	222.56	73.81	37.50	719.49

Table A.1: Computational time and flows are shown for 600 s simulations with tags OD (orthogonal decoupled) and TW (time window), respectively, with $N_p = \{6,\ 12\}$ horizons in the 8-loop. The test cases are: [-] without added constraints; [a] with added binary causality constraints; [b] with car-following related added binary constraints; [ab] with both types of added binary constraints. The number of cars was 10 with $\rho = 50$ $\frac{\text{veh}}{\text{km}}$

Type ($\rho = 100$ $[\frac{\text{veh}}{\text{km}}]$)	Binary constraints	Horizon	Computation time t_{opt} [ms]				Q $[\frac{\text{veh}}{\text{h}}]$
			Mean	Max.	Min.	Std.	
OD	-	$N_p = 6$	13.08	37.63	8.32	6.28	552.57
	a	$N_p = 6$	11.43	63.80	6.17	6.55	480.59
	b	$N_p = 6$	13.21	42.87	8.05	6.61	552.57
	ab	$N_p = 6$	11.56	54.68	6.39	6.58	480.59
TW	-	$N_p = 6$	13.31	37.26	7.79	5.09	553.14
	a	$N_p = 6$	14.69	35.10	7.50	5.49	553.14
	b	$N_p = 6$	14.10	41.42	7.84	5.76	553.14
	ab	$N_p = 6$	14.76	37.50	7.61	5.62	553.14
OD	-	$N_p = 12$	145.01	251.07	75.02	29.93	598.77
	a	$N_p = 12$	160.65	294.61	74.46	41.70	598.81
	b	$N_p = 12$	152.51	254.33	80.19	31.01	598.81
	ab	$N_p = 12$	164.51	310.32	74.55	44.97	598.81
TW	-	$N_p = 12$	146.88	271.55	74.48	33.73	608.78
	a	$N_p = 12$	161.25	320.62	59.85	44.61	605.03
	b	$N_p = 12$	148.04	266.54	80.64	30.74	608.78
	ab	$N_p = 12$	157.80	291.35	70.18	42.93	605.80

Table A.2: Computational time and flows are shown for 600 s simulations with tags OD (orthogonal decoupled) and TW (time window), respectively, with $N_p = \{6,\ 12\}$ horizons in the 8-loop. The test cases are: [-] without added constraints; [a] with added binary causality constraints; [b] with car-following related added binary constraints; [ab] with both types of added binary constraints. The number of cars was 10 with $\rho = 100$ $\frac{\text{veh}}{\text{km}}$

Type ($\rho = 120$ $\left[\frac{\text{veh}}{\text{km}}\right]$)	Binary constraints	Horizon	Computation time t_{opt} [ms]				Q $\left[\frac{\text{veh}}{\text{h}}\right]$
			Mean	Max.	Min.	Std.	
OD	-	$N_p = 6$	13.51	47.03	7.75	6.12	378.76
	a	$N_p = 6$	10.13	60.66	6.15	9.01	342.42
	b	$N_p = 6$	13.76	42.42	7.87	6.20	378.76
	ab	$N_p = 6$	10.24	55.25	6.27	8.96	342.42
TW	-	$N_p = 6$	17.11	44.59	7.84	7.99	399.30
	a	$N_p = 6$	15.31	42.73	6.59	6.60	399.30
	b	$N_p = 6$	16.27	50.99	7.83	7.49	399.30
	ab	$N_p = 6$	15.35	42.14	6.66	6.50	399.30
OD	-	$N_p = 12$	130.44	246.08	63.57	33.62	463.40
	a	$N_p = 12$	133.03	327.05	11.22	53.07	457.05
	b	$N_p = 12$	138.08	238.83	68.92	34.96	463.78
	ab	$N_p = 12$	135.21	318.77	11.60	51.86	459.73
TW	-	$N_p = 12$	137.15	239.73	68.64	33.50	463.76
	a	$N_p = 12$	142.09	272.48	57.66	45.31	463.76
	b	$N_p = 12$	139.22	252.23	55.02	38.52	487.12
	ab	$N_p = 12$	139.62	266.52	59.32	48.57	463.76

Table A.3: Computational time and flows are shown for 600 s simulations with tags OD (orthogonal decoupled) and TW (time window), respectively, with $N_p = \{6,\ 12\}$ horizons in the 8-loop. The test cases are: [-] without added constraints; [a] with added binary causality constraints; [b] with car-following related added binary constraints; [ab] with both types of added binary constraints. The number of cars was 10 with $\rho = 120$ $\frac{\text{veh}}{\text{km}}$

B

Appendix: Road inlet flow generation

For traffic simulation purposes a vehicle arrival model can be used to generate the traffic flow at road inlets.

This appendix aims to enumerate the Poisson distribution, as well as the closely related negative exponential distribution and its relevant measures [10], to then derive the general truncated exponential distribution for vehicle flow at the junction inlets. By using the design parameters of the truncated exponential distribution, random numbers are generated via the quantile function sampling.

Definitions and relations that are repeatedly used in the derivations are collected first [10].

Definition of expected value, mean, or first moment of a continuous random variable X:

$$\mathrm{E}\left[X\right] = \int_{-\infty}^{\infty} x \ f\left(x\right) \ dx,$$

where the Probability Density Function (PDF) is f of the hypothetical outcome x. In the case where X is a discrete random variable, the expected value is:

$$\mathrm{E}\left[X\right] = \sum_{i=1}^{\infty} x_i \ f(x_i),$$

with the probability mass function f of discrete hypothetical outcomes x_i.

The variance or second central moment (standard deviation squared σ^2) of a random variable X is as follows:

$$\mathrm{Var}\left[X\right] = \mathrm{E}\left[\left(X - \mathrm{E}\left[X\right]\right)^2\right] = E\left[X^2\right] - \mathrm{E}\left[X\right]^2. \tag{B.1}$$

Below, some frequent integrals and their solutions are shown; first:

$$\int e^{\alpha x} \ dx = \frac{1}{\alpha} e^{\alpha x}. \tag{B.2}$$

The next integral is:

$$\int xe^{\alpha x}\ dx,$$

which is solved with integration by parts:

$$\int u\left(x\right)\ v'\left(x\right)\ dx = u\left(x\right)v\left(x\right) - \int v\left(x\right)\ u\left(x\right)\ dx$$

with substitution of:

$$\begin{aligned} u\left(x\right) &= x; \quad & u'\left(x\right) &= 1 \\ v'\left(x\right) &= e^{\alpha x}; \quad & v\left(x\right) &= \int v'\left(x\right)dx = \frac{1}{\alpha}e^{\alpha x} \end{aligned}$$

gives:

$$\begin{aligned} \int xe^{\alpha x}\ dx &= x\frac{1}{\alpha}e^{\alpha x} - \int \frac{1}{a}e^{\alpha x}\ \ dx \\ &= \left(\frac{x}{\alpha} - \frac{1}{\alpha^2}\right)e^{\alpha x}. \end{aligned} \tag{B.3}$$

The following integral is:

$$\int x^2e^{\alpha x}\ dx,$$

which similarly can be solved using integration by parts, with substitution of:

$$\begin{aligned} u\left(x\right) &= x; \quad & u'\left(x\right) &= 1 \\ v'\left(x\right) &= xe^{\alpha x}; & v\left(x\right) &= \int v'\left(x\right)dx = \left(\frac{x}{\alpha} - \frac{1}{\alpha^2}\right)e^{\alpha x} \end{aligned}$$

yields:

$$\begin{aligned} \int x^2e^{\alpha x}\ dx &= x\left(\frac{x}{\alpha} - \frac{1}{\alpha^2}\right)e^{\alpha x} - \int \left(\frac{x}{\alpha} - \frac{1}{\alpha^2}\right)e^{\alpha x}dx \\ &= x\left(\frac{x}{\alpha} - \frac{1}{\alpha^2}\right)e^{\alpha x} - \frac{1}{\alpha}\left(\frac{x}{\alpha} - \frac{1}{\alpha^2}\right)e^{\alpha x} + \frac{1}{\alpha^3}e^{\alpha x}. \end{aligned}$$

In summary:

$$\int x^2e^{\alpha x}\ dx = \left(\frac{x^2}{\alpha} - \frac{2x}{\alpha^2} + \frac{2}{\alpha^3}\right)e^{\alpha x}. \tag{B.4}$$

Poisson distribution describes the probability of discrete events observed in identical consecutive intervals (i.e. the number of vehicles arriving at a fixed point in the road

within a defined time period).

The probability mass function is defined as:

$$f\left(x_i\right) = \frac{\lambda^{x_i} e^{-\lambda}}{x_i!},$$

for the number of hypothetical discrete events $x_i = 0, 1, 2 \ldots$ (number of vehicles arriving) which occurs with the expected rate value (average number of arrivings) $\lambda > 0$ over the interval.

Expected value (mean) and variance properties of the Poisson distribution with X discrete random variable:

$$\begin{aligned}\mathrm{E}\left[X\right] &= \sum_{x=0}^{\infty} x \; f(x) = \sum_{x=0}^{\infty} x \; \frac{\lambda^x e^{-\lambda}}{x!} = \sum_{x=1}^{\infty} x \; \frac{\lambda^x e^{-\lambda}}{x!} = \lambda e^{-\lambda} \sum_{x=1}^{\infty} \frac{\lambda^{x-1}}{(x-1)!} \\ &= \lambda e^{-\lambda} \sum_{x=0}^{\infty} \frac{\lambda^x}{x!} = \lambda e^{-\lambda} e = \lambda.\end{aligned}$$

Using the following expected value identity:

$$E[X + Y] = \mathrm{E}\left[X\right] + \mathrm{E}\left[Y\right]$$

gives:

$$\mathrm{E}\left[X^2\right] = \mathrm{E}\left[X\left(X-1\right) + X\right] = \mathrm{E}\left[X\left(X-1\right)\right] + \mathrm{E}\left[X\right],$$

where

$$\begin{aligned}\mathrm{E}\left[X\left(X-1\right)\right] &= \sum_{x=0}^{\infty} x\left(x-1\right) \; f(x) = \sum_{x=2}^{\infty} \; x\left(x-1\right) \; \frac{\lambda^x e^{-\lambda}}{x!} = \lambda^2 e^{-\lambda} \sum_{x=2}^{\infty} \; \frac{\lambda^{x-2}}{(x-2)!} \\ &= \lambda^2 e^{-\lambda} e^{\lambda} = \lambda^2.\end{aligned}$$

Thus, by substituting back the above results into (B.1), the variance yields:

$$\mathrm{Var}\left[X\right] = \mathrm{E}\left[X^2\right] - \mathrm{E}\left[X\right]^2 = \mathrm{E}\left[X\left(X-1\right)\right] + \mathrm{E}\left[X\right] - \mathrm{E}\left[X\right]^2 = \lambda.$$

In summary:

$$\begin{aligned}\mathrm{E}\left[X\right] &= \lambda \\ \mathrm{Var}\left[X\right] &= \lambda.\end{aligned}$$

One may be more inclined to obtain the distribution of Δt time gaps between the discrete events (continuous measure) rather than the number of events over a fixed interval. This formulation of the Poisson process is known as exponential distribution with the

probability distribution function:

$$f\left(y\right) = \lambda e^{-\lambda y},$$

where the rate parameter is $\lambda > 0$ and y is defined over $[0, \infty)$ interval. Since the PDF of exponential distribution is continuous, for Y exponentially distributed random variable the mean and variance over the semi-infinite interval:

$$\mathrm{E}\left[Y\right] = \int_0^\infty y\ f(y)\ dy = \int_0^\infty y\ \lambda e^{-\lambda y}\ dy.$$

This expression can be solved using the integration-by-parts identity:

$$\int u\left(x\right)\ v'\left(x\right)\ dx = u\left(x\right)v\left(x\right) - \int v\left(x\right)\ u\left(x\right)\ dx$$

with a substitution of:

$$\begin{aligned} u\left(x\right) &= x; \quad & u'\left(x\right) &= 1 \\ v'\left(x\right) &= e^{\alpha x}; \quad & v\left(x\right) &= \int v'\left(x\right)dx = \frac{1}{\alpha}e^{\alpha x} \end{aligned}$$

reaching:

$$\int xe^{\alpha x}\ dx = x\frac{1}{\alpha}e^{\alpha x} - \int \frac{1}{a}e^{\alpha x}\ dx = \left(\frac{x}{\alpha} - \frac{1}{\alpha^2}\right)e^{\alpha x}.$$

Then

$$\mathrm{E}\left[Y\right] = \int_0^\infty y\ \lambda e^{-\lambda y}\ dy = \lambda\left[\left(-\frac{y}{\lambda} - \frac{1}{\lambda^2}\right)e^{-\lambda y}\right]_0^\infty = \lambda^{-1}.$$

In case of the variance:

$$\mathrm{Var}\left[Y\right] = \int_0^\infty y^2 f(y)d\tau - \mathrm{E}\left[Y\right]^2 = \int_0^\infty y^2\lambda e^{-\lambda\tau}d\tau - \lambda^{-2}.$$

Finally, using the integral (B.4) and the expected value sub-results from above:

$$\begin{aligned} \mathrm{Var}\left[Y\right] &= \int_0^\infty y^2\lambda e^{-\lambda\tau}d\tau - \lambda^{-2} = \left[\left(y^2 - \frac{2y}{\lambda} - \frac{2}{\lambda^2}\right)e^{-\lambda y}\right]_0^\infty - \lambda^{-2} \\ &= 2\lambda^{-2} - \lambda^{-2} = \lambda^{-2}. \end{aligned}$$

In summary:

$$\begin{aligned} \mathrm{E}\left[Y\right] &= \lambda^{-1} \\ \mathrm{Var}\left[Y\right] &= \lambda^{-2}. \end{aligned}$$

Remark: The inlet can now generate vehicle-arrival time gaps according to an exponential distribution over the interval of $[0, \infty)$. However, with relatively low probability, very high following times could occur, resulting in a rather uneventful simulation. On the other hand, generating vehicle arrival times close to each other would mean that a vehicle cannot yet physically leave the space on the road before the next agent would have to appear in that same spot. This phenomenon remains a disadvantageous characteristic of the inlet with a hard demand on vehicles appearing, whereby adding a lower bound on the following times decreases the number of hard exceptions to be treated.

For more general and flexible settings on the inlet mechanism, a lower- and upper-truncated exponential distribution is formulated with a settable mean. This distribution is later sampled to create Z continuous random variables to express vehicle headways at the inlet.

$$f\left(z\right) := \begin{cases} \frac{\varphi}{\psi} e^{\varphi z}, & a < Z \leq b \\ 0, & \text{otherwise} \end{cases}$$

where the lower limit is a and the upper limit of truncation is b, moreover, $b - a > 0$; the parameters defining the general shape of the PDF are φ and ψ.

The Cumulative Density Function (CDF) is calculated from the PDF by:

$$F(z) = \int_{-\infty}^{z} f\left(\hat{z}\right) \, d\hat{z}.$$

Moreover, the PDF is chosen to give:

$$F\left(a\right) = 0 \tag{B.5}$$

$$F\left(b\right) = 1. \tag{B.6}$$

One more design element is the mean value ω which must lie within the boundaries $a < \omega < b$:

$$\mathrm{E}\left[Z\right] = \int_{a}^{b} z \; f\left(z\right) \, dz = \omega$$

First, (B.5) and (B.6) conditions are met by:

$$\int_{a}^{b} f\left(z\right) \, dz = 1$$

$$\int_{a}^{b} \frac{\varphi}{\psi} e^{\varphi z} \, dz = \frac{1}{\psi} \left[e^{\varphi z}\right]_{a}^{b} = \frac{\left(e^{\varphi b} - e^{\varphi a}\right)}{\psi} = 1$$

$$\psi = e^{\varphi b} - e^{\varphi a}. \tag{B.7}$$

Then, the condition for the mean is used:

$$\begin{aligned}\omega &= \int_a^b z\ f(z)\ dz = \int_a^b z\ \frac{\varphi}{\psi} e^{\varphi z}\ dz = \frac{1}{\psi}\left[\left(z - \frac{1}{\varphi}\right) e^{\varphi z}\right]_a^b \\ &= \frac{1}{\psi}\left(b - \frac{1}{\varphi}\right) e^{\varphi b} - \frac{1}{\psi}\left(a - \frac{1}{\varphi}\right) e^{\varphi a}.\end{aligned}$$

Substituting the results from (B.7) gives:

$$\omega = b + \frac{(b-a)}{e^{\varphi(b-a)} - 1} - \frac{1}{\varphi}$$

which contains φ in an implicit manner and can be solved with a generic numerical solver (e.g. fzero in Matlab). This truncation is formulated only for practical use; better variable choices and more elegant formulae can be obtained.

The derived truncated distribution can describe not only truncated negative exponential distributions but also positive exponential distributions, which would apply, for example, if vehicles were tend to have longer following times more often than short ones. In traffic flows, however, this is not a characteristic trend; in fact, vehicles tend to have short following times more often, resulting in negative exponential trends.

B.1 Sampling the truncated exponential distribution

Recall the CDF:

$$F(z) = \int_a^z f(\hat{z})\ d\hat{z} = \int_a^z \frac{\varphi}{\psi} e^{\varphi \hat{z}} d\hat{z} = \frac{e^{\varphi z} - e^{\varphi a}}{\psi}.$$

Thus, the quantile function is:

$$F^{-1}(p) = \frac{\ln(\psi p + e^{\varphi a})}{\varphi}, \qquad 0 \le p < 1$$

where the probability is p.

After substituting p element wise from a vector of uniformly distributed random numbers, with elements on a $[0, 1)$ interval, a sampled, truncated, exponentially distributed random number vector can be generated for the vehicle headway times.

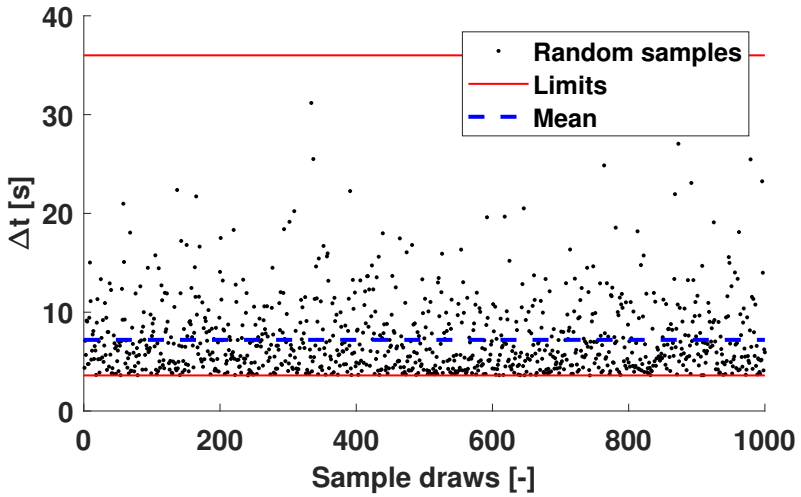


Figure B.1: Random Δt vehicle following times drawn from a distribution

B.2 Sample example

The truncated exponential inlet distribution can be defined with three parameters, using flows:

$$Q_{\text{min}} = 100 \ \tfrac{\text{veh}}{\text{h}}$$
$$Q_{\text{mean}} = 500 \ \tfrac{\text{veh}}{\text{h}}$$
$$Q_{\text{max}} = 1000 \ \tfrac{\text{veh}}{\text{h}}$$

or, alternatively, Δt vehicle following times:

$$\Delta t_{\text{min}} = Q_{\text{max}}^{-1} = 3.6 \text{ s}$$
$$\Delta t_{mean} = Q_{\text{mean}}^{-1} = 7.2 \text{ s}$$
$$\Delta t_{\text{max}} = Q_{\text{min}}^{-1} = 36 \text{ s}$$

giving problem specific variables $a = \Delta t_{\text{min}}$, $b = \Delta t_{\text{max}}$ and $\omega = \Delta t_{\text{mean}}$, then calculating $\varphi = 0.2775$ and $\psi = 0.3682$, which define the distribution parameters.

As an exercise, 1000 samples are drawn from the distribution specified above (shown on Figure B.1), which result in the mean $\text{E}\left[\Delta t\right] = 7.383$ s and $\text{Var}\left[\Delta t\right] = 3.8694$ s.

Vehicle-arrival times can then be calculated by cumulatively summing the random Δt time gaps sampled from the distribution and repeating this for other road inlets which generally have different distributions.

Bibliography

[1] Altche, F.: 2018, *Decision-based motion planning for cooperative and autonomous vehicles*, Theses, PSL Research University.
URL: *https://pastel.archives-ouvertes.fr/tel-02073593*

[2] Altche, F., Qian, X. and de La Fortelle, A.: 2016, Time-optimal coordination of mobile robots along specified paths, *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE.

[3] Altche, F., Qian, X. and de La Fortelle, A.: 2017, An Algorithm for Supervised Driving of Cooperative Semi-Autonomous Vehicles, *IEEE Transactions on Intelligent Transportation Systems* **18**(12), 3527–3539.

[4] Athans, M.: 1969, A unified approach to the vehicle-merging problem, *Transportation Research* **3**(1), 123–133.

[5] Bali, C. and Richards, A.: 2017, Robot navigation using convex model predictive control and approximate operating region optimization, *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp. 2171–2176.

[6] Bali, C. and Richards, A.: 2018, Merging Vehicles at Junctions using Mixed-Integer Model Predictive Control, *2018 European Control Conference (ECC)*, IEEE, pp. 1740–1745.

[7] *BBC: Tesla Model 3: Autopilot engaged during fatal crash*: 2019, https://www.bbc.co.uk/news/technology-48308852.

[8] Bemporad, A. and Morari, M.: 1999, Control of Systems integrating Logic Dynamics and Constraints, *Automatica* **35(3)**, 407–427.

[9] Blanchini, F.: 1999, Set invariance in control, *Automatica* **35**(11), 1747–1767.

[10] Blitzstein, J. K. and Hwang, J.: 2019, *Introduction to Probability, Second Edition*, Taylor & Francis Ltd.

[11] Borrelli, F., Bemporad, A. and Morari, M.: 2017, *Predictive Control for Linear and Hybrid Systems*, Cambridge University Press.

[12] Campbell, M., Egerstedt, M., How, J. P. and Murray, R. M.: 2010, Autonomous driving in urban environments: approaches, lessons and challenges, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **368**(1928), 4649–4672.

[13] Carson III, J. M.: 2008, *Robust model predictive control with a reactive safety mode*, PhD thesis, California Institute of Technology.

[14] Chen, L. and Englund, C.: 2016, Cooperative intersection management: A survey, *IEEE Transactions on Intelligent Transportation Systems* **17**(2), 570–586.

[15] Csorvasi, G. and Vajk, I.: 2016, Analysis of an on-line minimum-time velocity optimization algorithm, *2016 17th International Carpathian Control Conference (ICCC)*, IEEE, pp. 122–127.

[16] CVX Research, Inc.: 2012, CVX: Matlab software for disciplined convex programming, version 2.0, `http://cvxr.com/cvx`.

[17] Dahl, J., de Campos, G. R., Olsson, C. and Fredriksson, J.: 2019, Collision avoidance: A literature review on threat-assessment techniques, *IEEE Transactions on Intelligent Vehicles* **4**(1), 101–113.

[18] de Campos, G. R., Falcone, P., Hult, R., Wymeersch, H. and Sjoberg, J.: 2017, Traffic coordination at road intersections: Autonomous decision-making algorithms using model-based heuristics, *IEEE Intelligent Transportation Systems Magazine* **9**(1), 8–21.

[19] de Campos, G. R., Falcone, P. and Sjoberg, J.: 2013, Autonomous cooperative driving: A velocity-based negotiation approach for intersection crossing, *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, IEEE.

[20] de La Fortelle, A.: 2010, Analysis of reservation algorithms for cooperative planning at intersections, *13th International IEEE Conference on Intelligent Transportation Systems*, IEEE.

[21] Debada, E., Makarem, L. and Gillet, D.: 2017, A virtual vehicle based coordination framework for autonomous vehicles in heterogeneous scenarios, *2017 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, IEEE.

[22] Dresner, K. and Stone, P.: 2004, Multiagent traffic management: A reservation-based intersection control mechanism, *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '04, IEEE Computer Society, Washington, DC, USA, pp. 530–537.

[23] Dunbar, W. B. and Murray, R. M.: 2006, Distributed receding horizon control for multi-vehicle formation stabilization, *Automatica* **42**(4), 549–558.

[24] Eele, A. and Richards, A. G.: 2007, Path-planning with avoidance using nonlinear branch-and-bound optimisation, *AIAA Guidance, Navigation and Control Conference*.

[25] ERSO: 2018, Traffic safety basic facts (latest data: 2016), online. European Road Safety Observatory (ERSO).

[26] Fankhauser, B., Makarem, L. and Gillet, D.: 2011, Collision-free intersection crossing of mobile robots using decentralized navigation functions on predefined paths, *2011 IEEE 5th International Conference on Cybernetics and Intelligent Systems (CIS)*, IEEE.

[27] Flores, C., Milanes, V. and Nashashibi, F.: 2017, A time gap-based spacing policy for full-range car-following, *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE.

[28] Grant, M. and Boyd, S.: 2008, Graph implementations for nonsmooth convex programs, *in* V. Blondel, S. Boyd and H. Kimura (eds), *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, Springer-Verlag Limited, pp. 95–110.

[29] Greatwood, C. and Richards, A. G.: 2019, Reinforcement learning and model predictive control for robust embedded quadrotor guidance and control, *Autonomous Robots* **43**(7), 1681–1693.

[30] Gregoire, J.: 2014, *Priority-based coordination of mobile robots*, PhD thesis, Mines ParisTech.

[31] Gregoire, J., Bonnabel, S. and de La Fortelle, A.: 2013, Robust multirobot coordination using priority encoded homotopic constraints, p. 21.

[32] Gurobi Optimization, LLC: 2019, Gurobi Optimizer Reference Manual.
URL: *http://www.gurobi.com*

[33] Haklay, M. and Weber, P.: 2008, OpenStreetMap: User-generated street maps, *IEEE Pervasive Computing* **7**(4), 12–18.

[34] Hult, R., Zanon, M., Gras, S. and Falcone, P.: 2018, An MIQP-based heuristic for Optimal Coordination of Vehicles at Intersections, *2018 IEEE Conference on Decision and Control (CDC)*, IEEE, pp. 2783–2790.

[35] Hult, R., Zanon, M., Gros, S. and Falcone, P.: 2016, Primal decomposition of the optimal coordination of vehicles at traffic intersections, *2016 IEEE 55th Conference on Decision and Control (CDC)*, IEEE.

[36] Hult, R., Zanon, M., Gros, S., Wymeersch, H. and Falcone, P.: 2019, Optimization-based coordination of connected, automated vehicles at intersections. Vehicle System Dynamics.

[37] *INRIX 2018 Global Traffic Scorecard*: 2019, online.
URL: *http://inrix.com/press-releases/scorecard-2018-uk/*

[38] Jung, M.: 2014, Relaxations and approximations for mixed-integer optimal control.

[39] Kamal, M. A. S., ichi Imura, J., Hayakawa, T., Ohata, A. and Aihara, K.: 2015, A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights, *IEEE Transactions on Intelligent Transportation Systems* **16**(3), 1136–1147.

[40] Kant, K. and Zucker, S. W.: 1986, Toward efficient trajectory planning: The path-velocity decomposition, *The International Journal of Robotics Research* **5**(3), 72–89.

[41] Kerrigan, E. C. and Maciejowski, J. M.: 2000, Soft Constraints And Exact Penalty Functions In Model Predictive Control, *Control 2000 Conference* .

[42] Kerrigan, E. and Maciejowski, J.: 2001, Robust feasibility in model predictive control: necessary and sufficient conditions, *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No.01CH37228)*, Vol. 1, IEEE, pp. 728–733.

[43] Kim, K.-D. and Kumar, P. R.: 2014, An MPC-based approach to provable system-wide safety and liveness of autonomous ground traffic, *IEEE Transactions on Automatic Control* **59**(12), 3341–3356.

[44] Kowshik, H., Caveney, D. and Kumar, P. R.: 2011, Provable systemwide safety in intelligent intersections, *IEEE Transactions on Vehicular Technology* **60**(3), 804–818.

[45] LaValle, S.: 2006, *Planning Algorithms*, Cambridge University Press.

[46] Levinson, J., Askeland, J., Becker, J., Dolson, J., Held, D., Kammel, S., Kolter, J. Z., Langer, D., Pink, O., Pratt, V., Sokolsky, M., Stanek, G., Stavens, D., Teichman, A., Werling, M. and Thrun, S.: 2011, Towards fully autonomous driving: Systems and algorithms, *2011 IEEE Intelligent Vehicles Symposium (IV)*, IEEE.

[47] Luo, L., Chen, J. and Zhang, F.: 2016, Integrated adaptive cruise control design considering the optimization of switching between throttle and brake, *2016 IEEE Intelligent Vehicles Symposium (IV)*, IEEE.

[48] Maciejowski, J.: 2000, *Predictive Control with Constraints*, Prentice Hall.

[49] Maestre, J. M. and Negenborn, R. R. (eds): 2014, *Distributed Model Predictive Control Made Easy*, Springer Netherlands.

[50] Maia, M. H. and Galvão, R. K. H.: 2009, On the use of mixed-integer linear programming for predictive control with avoidance constraints, *International Journal of Robust and Nonlinear Control* **19**(7), 822–828.

[51] Makarem, L. and Gillet, D.: 2012, Fluent coordination of autonomous vehicles at intersections, *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE.

[52] Makarem, L. and Gillet, D.: 2013, Model predictive coordination of autonomous vehicles crossing intersections, *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, IEEE.

[53] Mayne, D. Q., Rawlings, J. B., Rao, C. V. and Scokaert, P. O. M.: 2000, Constrained Model Predictive Control: Stability and Optimality, *Automatica* **36**, 789–814.

[54] Paden, B., Cap, M., Yong, S. Z., Yershov, D. and Frazzoli, E.: 2016, A survey of motion planning and control techniques for self-driving urban vehicles, *IEEE Transactions on Intelligent Vehicles* **1**(1), 33–55.

[55] Pande, A.: 2015, *Traffic engineering handbook*, John Wiley & Sons Inc, Hoboken, New Jersey.

[56] Prodan, I., Stoican, F., Olaru, S. and Niculescu, S.-I.: 2016, *Mixed-Integer Representations in Control Design*, Springer International Publishing.

[57] Qian, X.: 2016, *Model predictive control for autonomous and cooperative driving*, Theses, PSL Research University.
URL: *https://pastel.archives-ouvertes.fr/tel-01635261*

[58] Qian, X., Gregoire, J., de La Fortelle, A. and Moutarde, F.: 2015, Decentralized model predictive control for smooth coordination of automated vehicles at intersection, *2015 European Control Conference (ECC)*, IEEE, pp. 3452–3458.

[59] Rawlings, J. B., Mayne, D. Q. and Diehl, M. M.: 2017, *Model Predictive Control: Theory, Computation, and Design, 2nd Edition*, Nob Hill Publishing, LLC.

[60] Richards, A. G.: 2002, *Trajectory control using mixed integer linear programming*, Master's thesis, MIT.

[61] Richards, A. G. and How, J. P.: 2005, Mixed-integer Programming for Control, *Proceedings of the American Control Conference*, IEEE, Portland, Oregon, pp. 2676–2683.

[62] Richards, A. and How, J. P.: 2007, Robust distributed model predictive control, *International Journal of Control* **80**(9), 1517–1531.

[63] Richards, A. and Turnbull, O.: 2013, Inter-sample avoidance in trajectory optimizers using mixed-integer linear programming, *International Journal of Robust and Nonlinear Control* **25**(4), 521–526.

[64] Rios-Torres, J. and Malikopoulos, A. A.: 2017a, Automated and cooperative vehicle merging at highway on-ramps, *IEEE Transactions on Intelligent Transportation Systems* **18**(4), 780–789.

[65] Rios-Torres, J. and Malikopoulos, A. A.: 2017b, A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps, *IEEE Transactions on Intelligent Transportation Systems* **18**(5), 1066–1077.

[66] Rizaldi, A. and Althoff, M.: 2015, Formalising traffic rules for accountability of autonomous vehicles, *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, IEEE.

[67] Rosolia, U., Bruyne, S. D. and Alleyne, A. G.: 2017, Autonomous vehicle control: A non-convex approach for obstacle avoidance, *IEEE Transactions on Control Systems Technology* **25**(2), 469–484.

[68] Sathya, A., Sopasakis, P., Parys, R. V., Themelis, A., Pipeleers, G. and Patrinos, P.: 2018, Embedded nonlinear model predictive control for obstacle avoidance using PANOC, *2018 European Control Conference (ECC)*, IEEE.

[69] Scattolini, R.: 2009, Architectures for distributed and hierarchical Model Predictive Control – A review, *Journal of Process Control* **19**(5), 723–731.

[70] Schouwenaars, T., Moor, B. D., Feron, E. and How, J.: 2001, Mixed integer programming for multi-vehicle path planning, *2001 European Control Conference (ECC)*, IEEE.

[71] Shi, J., Zheng, Y., Jiang, Y., Zanon, M., Hult, R. and Houskal, B.: 2018, Distributed control algorithm for vehicle coordination at traffic intersections, *2018 European Control Conference (ECC)*, IEEE, pp. 1166–1171.

[72] Shin, Y. and Bien, Z.: 1989, Collision–Free Trajectory Planning for Two Robot Arms, *Robotica* **7**(03), 205.

[73] Singh, S.: 2018, Critical reasons for crashes investigated in the national motor vehicle crash causation survey, *Technical Report DOT HS 812 506*, National Highway Traffic Safety Administration, Washington, DC.

[74] Thrun, S.: 2010, Toward robotic cars, *Communications of the ACM* **53**(4), 99.

[75] Treiber, M. and Kesting, A.: 2013, *Traffic Flow Dynamics*, Springer Berlin Heidelberg.

[76] Trodden, P. A. and Richards, A. G.: 2013, Cooperative tube-based distributed MPC for linear uncertain systems coupled via constraints, *Intelligent Systems, Control and Automation: Science and Engineering, (Distributed Model Predictive Control Made Easy)*, Springer Netherlands, pp. 57–72.

[77] UK Department for Transport: 2015, The Highway Code, available at https://www.gov.uk/guidance/the-highway-code.

[78] Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M. N., Dolan, J., Duggins, D., Galatali, T., Geyer, C., Gittleman, M., Harbaugh, S., Hebert, M., Howard, T. M., Kolski, S., Kelly, A., Likhachev, M., McNaughton, M., Miller, N., Peterson, K., Pilnick, B., Rajkumar, R., Rybski, P., Salesky, B., Seo, Y.-W., Singh, S., Snider, J., Stentz, A., Whittaker, W. "., Wolkowicki, Z., Ziglar, J., Bae, H., Brown, T., Demitrish, D., Litkouhi, B., Nickolaou, J., Sadekar, V., Zhang, W., Struble, J., Taylor, M., Darms, M. and Ferguson, D.: 2008, Autonomous driving in urban environments: Boss and the Urban Challenge, *Journal of Field Robotics* **25**(8), 425–466.

[79] Velenis, E. and Tsiotras, P.: 2008, Minimum-time travel for a vehicle with acceleration limits: Theoretical analysis and receding-horizon implementation, *Journal of Optimization Theory and Applications* **138**(2), 275–296.

[80] Wakabayashi, D.: 2018, Self-driving uber car kills pedestrian in arizona, where robots roam, The New York Times, online.
URL: *https://www.nytimes.com/2018/03/19/technology/uber-driverless-fatality.html*

[81] Wang, Y. and Boyd, S.: 2010, Fast model predictive control using online optimization, *IEEE Transactions on Control Systems Technology* **18**(2), 267–278.

[82] Williams, H. P. and Brailsford, S. C.: 1996, *Advances in Linear and Integer Programming*, Clarendon Press, Oxford, UK, chapter Computational Logic and Integer Programming, pp. 249–281.

[83] Xu, S. and Peng, H.: 2018, Design and comparison of fuel-saving speed planning algorithms for automated vehicles, *IEEE Access* **6**, 9070–9080.

[84] Yi, B., Gottschling, S., Ferdinand, J., Simm, N., Bonarens, F. and Stiller, C.: 2016, Real time integrated vehicle dynamics control and trajectory planning with MPC for critical maneuvers, *2016 IEEE Intelligent Vehicles Symposium (IV)*, IEEE.

[85] Zhang, Y. J., Malikopoulos, A. A. and Cassandras, C. G.: 2016, Optimal control and coordination of connected and automated vehicles at urban traffic intersections, *2016 American Control Conference (ACC)*, IEEE.