Doctoral Thesis

Lane-Free Crossing of Connected and Autonomous Vehicles through Intersections

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the

Department of Engineering and Informatics

by

Mahdi Amouzadi

March, 2023

Abstract

Unlike conventional cars, connected and autonomous vehicles (CAVs) can cross intersections in a lane-free order. Lane-free crossing allows CAVs to make use of all segments of roads and intersections which is hypothesised to improve energy consumption and traffic throughput. However, controlling CAVs to pass through intersections in a lane-free order is a challenging minimum-time optimal control problem (OCP) with non-differentiable collision avoidance constraints and is difficult to be formulated and solved. This thesis addresses this challenge by proposing: i) a control strategy that incorporates differentiable collision avoidance constraints to cross CAVs through lane-free intersections; ii) a framework to evaluate the capacity of intersections when CAVs are crossing in a lane-free order.

The proposed control strategy is formulated as a multi-objective OCP that minimises a combination of the crossing time and the energy consumption of CAVs due to their accelerations. The non-differentiable constraints that avoid collisions of vehicles with each other and with road boundaries are smoothed by applying the dual problem theory. It is shown that the solution of the formulated OCP when the crossing time is the only objective provides a lower bound of the crossing time of a junction which is exceptionally close to the theoretical limit. The calculated lower bound is a feasible benchmark to evaluate the performance of other intersection crossing algorithms. Considering the energy consumption as well, the results show that the proposed lane-free strategy reduces the crossing time of vehicles by an average of 40% as compared to the state-of-the-art reservation-based method, whilst consuming the same amount of energy. Furthermore, it is shown that crossing time through a lane-free intersection is fixed to a constant value regardless of the number of the crossing CAVs.

This work also proposes a novel framework including a measure and an algorithm to quantify the capacity of the lane-free intersections. The available measures to assess capacity of the conventional intersections are not applicable to the lane-free ones because the conventional roads restrain vehicles to travel within lanes. The results of this thesis show that a lane-free crossing of CAVs increases the capacity of intersections by, respectively, 127% and 36% as compared to the signalised crossing by human-drivers and by CAVs. A sensitivity analysis indicates that, in contrast to the signalised intersections, the capacity of the lane-free ones is improved by an increase in the initial speed and the maximum permissible speed and acceleration of the vehicles.

Declaration

This thesis is carried out at the Engineering and Informatics department of University of Sussex. I hereby declare that this thesis has not been and will not be, submitted in whole or in part to another University for the award of any other degree. I also confirm that any contributions, ideas or opinions of other works is acknowledged in this thesis.

Signature:

Mahdi Amouzadi

Acknowledgements

First and foremost, I would like to express my deepest gratitude towards my principal supervisor, Dr Arash M. Dizqah whom supported me and been very patient with me all the way through. Without his timely advice and encouragements this thesis would not have been in this form. I have been very fortunate to have him as my supervisor and most importantly as a friend. For all of this, Arash, thank you!

I would also like to thank my second supervisor, Prof Julian Dunne for his support for achieving this scholarship. My PhD has been fully funded by the school of Engineering and Informatics at the University of Sussex and for that I am thankful.

My special thanks goes to my friends Dr Hamid Enayatollahi, Dr Alireza Sarmadian and Mr Amirhossein Naseredini who provided much support and suggestions during writing this thesis.

Last but not least, I would like to thank my family for their love, encouragements and support throughout my studies.

Contents

List of Tables

List of Figures

Nomenclature

$\mathcal{G}$	dual function
$\mathcal{H}$	Hamiltonian equation
$\mathcal{L}$	Lagrange interpolation
$\mathcal{O}$	polytope of road boundary r
$\mathcal{P}_i$	polytope of CAV i
$\mathscr{L}$	Lagrange function
$\mathbf{A}$	polytpoic orientation
$\mathbf{b}$	polytpoic size
$\mathbf{Q}$	gain of CAVs' pose and energy consumption
$\mathbf{s}$	separating hyperplane
a	acceleration (m/s^2)
C	capacity (veh/h)
C_F	front tyres cornering stiffness $(N/°)$
C_R	rear tyres cornering stiffness $(N/°)$
E_i	energy consumption of CAV i
g	equality constraint function
h	inequality constraint function
k	sampling step

L Stage cost of an OCP

l_f distance of the front axis from the centre of the gravity of the vehicle (m)

l_r distance of the rear axis from the centre of the gravity of the vehicle (m)

N_p number of control intervals

N_v number of vehicles (veh)

r yaw rate (rad/s)

T crossing time (s)

$T_{h,CAVs}$ headway for CAVs (s)

U permissible range of control inputs

V velocity (m/s)

X permissible range of system states

x non-inertial x position (m)

y non-inertial y position (m)

z_i pose of CAV i which includes x_i, y_i and θ_i

Abbreviations

AD automatic differentiation

$AWSC$ all-way stop-controlled

CAS computer algebra system

$CasADi$ computer algebra system with automatic differentiation implementation

$CAVs$ connected autonomous vehicles

CoG centre of gravity

$D-O$ discretise then optimise

DAE differential algebraic equation

DoF degree of freedom

DoU degree of utilisation

DP dynamic programming

HJB Hamilton-Jacobi-Bellman

HVs human-driven vehicles

IPM interior-point method

KKT Karush-Kuhn-Tucker

MPC model-predictive control

NLP nonlinear programming

OCP optimal control problem

ODE ordinary differential equation

PDE partial differential equation

PMP Pontryagin's minimum principle

SAE society of automotive engineers

SQP sequential quadratic programming

$TWSC$ two-way stop-controlled

$V2I$ vehicle-to-infrastructure

$V2V$ vehicle-to-vehicle

Subscripts

F front wheel

f front axis of a vehicle

i CAV number

p control interval

R rear wheel

r rear axis of a vehicle

v vehicle

Greek symbols

α crossing time gain

β sideslip angle (rad)

$\boldsymbol{\lambda}$ Lagrange variable

γ acceleration gain

μ barrier parameter

υ Lagrange variable

Chapter 1

Introduction

Connected and Autonomous Vehicles (CAVs), unlike conventional cars, are not restricted to drive within lanes and can travel in a lane-free order. Such lane-free movement allows CAVs to utilise the whole area of intersections which can be beneficial in terms of reducing the energy consumption and the travelling time and improving the traffic capacity of intersections. This thesis attempts at answering the following two research questions: 1) does lane-free crossing of CAVs through intersections reduce energy consumption and crossing time?; 2) does lane-free crossing of CAVs improve the capacity of intersections? This chapter provides the aims, objectives and contributions followed by the structure of the thesis.

1.1 Aims and Objectives

The aims of the research are to address the above-mentioned questions by:

- Design a control strategy for lane-free crossing of CAVs through intersections.
- Design a framework to measure the capacity of intersections when CAVs are crossing in a lane-free order.

The key objectives of this study are as follows:

- To formulate the lane-free crossing problem of CAVs through intersections as a minimum-time optimal control problem (OCP) with crossing time and the energy consumption as control objectives.
- To evaluate the minimum crossing time of CAVs through intersections in a lane-free order as compared to the state-of-the-art intersection crossing algorithms.

- To develop a measure to quantify the capacity of lane-free intersections.
- To quantify the capacity improvement of intersections by the lane-free crossing as compared to the current signalised crossing of HVs and CAVs.

1.2 Research Contributions

This study archives the following novel contributions:

- Formulation of the lane-free crossing of CAVs through intersections as a minimum-time OCP (published in [7, 8]);
- Smoothing of the constraints that avoid collisions of CAVs with each other and with road boundaries using the dual problem theory of convex optimisation. The original non-differentiable constraints are also relaxed with sufficient conditions to make them computationally inexpensive; (published in [7, 8]);
- Minimisation of the crossing time of multiple CAVs passing through intersections in a lane-free order. It is shown that the minimum crossing time calculated by the proposed algorithm is very close to its theoretical limit. The calculated optimal crossing time for a junction is shown to be fixed to a constant value regardless of the number of CAVs until reaching the maximum temporal-spatial capacity of the intersection (published in [7, 8]);
- Analysis and comparison of crossing time, energy consumption (due to acceleration) and passenger comfort of the proposed lane-free control strategy against the state-of-the-art reservation-based and lane-free methods. It is shown that the proposed lane-free strategy significantly improves the crossing time and passenger comfort while consuming the same amount of energy as the benchmark methods (published in [8]).
- A novel framework to quantify the capacity of lane-free intersections. The framework consists of a novel measure of capacity along with an algorithm to calculate this measure (published in [9]).
- Assessment of the efficacy of lane-free crossing to the capacity of intersections and compare the resulting traffic throughput as compared to the one by the signalised crossing of human-driven vehicles (HVs) and CAVs (published in [9]).

- A sensitivity analysis of the capacity and crossing time of the lane-free intersections with respect to the maximum permissible speed, maximum permissible acceleration/deceleration, initial speed and the number of the crossing vehicles (published in [9]).

1.3 Structure of the Thesis

The remainder of the thesis is structured as follows:

Chapter 2 provides a detailed background including literature review for the topic of this thesis. It starts by explaining different levels of automation and the technologies involved in CAVs followed by a discussion of path planning algorithms. In particular, it reviews the path planning algorithms for intersections including the corresponding challenges. Moreover, an example layout of a lane-free intersection is provided with a detailed explanation. The chapter lastly discusses the concept of optimisation of non-convex problems and explains the numerical methods for solving OCPs.

In chapter 3, the shape of vehicles and road boundaries are initially modelled and then the OCP of CAVs crossing intersections in a lane-free order is formulated. This chapter also compares the effectiveness of the proposed lane-free control strategy against a state-of-the-art reservation-based method and another lane-free method in terms of crossing time, energy consumption and passenger consumption.

Chapter 4 presents the framework including a measure and an algorithm that evaluates the capacity of lane-free intersections. The capacity of a lane-free intersection is provided and compared against the indicative Highway Capacity Manual (HCM) capacity of signalised intersections. The chapter also includes a sensitivity analysis of crossing time and capacity of lane-free intersections.

Chapter 5 provides a summery and draws the conclusion of this thesis followed by the direction of future works.

Chapter 2

Background

This chapter is dedicated to discuss the related background to the topic of this thesis. It begins with explaining how CAVs potentially improve the current transportation system and then presents a discussion on the design of CAVs including the hardware and software components, as well as the vehicle dynamics within the scope of this study. After reviewing several path planning techniques for CAVs, the problem of CAVs crossing intersections is introduced. In this regards, the-state-of-the-art algorithms and their corresponding methods for evaluating traffic capacity of intersections are reviewed. This chapter also describes the studied lane-free intersections with a graphical representation. As the lane-free crossing of CAVs through intersections is a non-convex minimum-time OCP, a detailed description of OCP and MPC is provided, followed by a discussion of optimisation problems. Finally, the chapter reviews the methods of solving OCPs and explains in detail the chosen solvers in this study.

2.1 Connected and Autonomous Vehicles

CAVs have become the recent hot topic for the researchers due to their abilities which is shown to be significantly effective in reducing energy consumption and travelling time while providing safety and comfort [10, 11]. These smart vehicles are able to communicate with each other and with infrastructure through Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) platforms and nullify wrong habits of human drivers -such as aggressive acceleration, riding the clutch, late braking and plenty of others- that improve fuel consumption and enhance the safety of the passengers [12, 13].

Moreover, CAVs can improve the current transportation system in terms of reducing traffic congestion. The traffic congestion of USA urban roads in 2014 forced people to spend extra 6.9 billion hours driving on the road and buy 3.1 billion gallons of fuel which resulted in an approximate cost of 160 billion USD [14]. In addition, with the current rate of increase in the population of cities, the demand for vehicles is significantly escalating while the capacity of roads is not changing which results in overloaded roads. CAVs can keep a shorter safety distance from each other than conventional human-driven vehicles (HVs) to resolve such issues with the current transportation system. Besides, CAVs can improve safely aspects of passengers and environmental features. The National Transportation Statistics of USA [15] states that vehicles in fifty states of America as well as Columbia caused 35,000 deaths and 2.2 million nonfatal harm, and nearly 1.7 billion metric tons of emission in form of CO_2 was freed to the environment in 2012. CAVs are autonomous vehicles that require less response time than HVs which can significantly enhance the economical and health issues raised above. However, in order to benefit the most from all these improvements it is important to analyse different levels of automated driving.

Figure 2.1 shows different levels of automated driving defined by the society of automotive engineers (SAE) [3]. Level 0 indicates that there is no automation system and it is related to the conventional vehicles where the human-driver performs all the tasks of driving. At level 1 of automation, vehicles are equipped with cruise control or even adaptive cruise control that allows vehicles to follow a preceding vehicle with a specified speed.

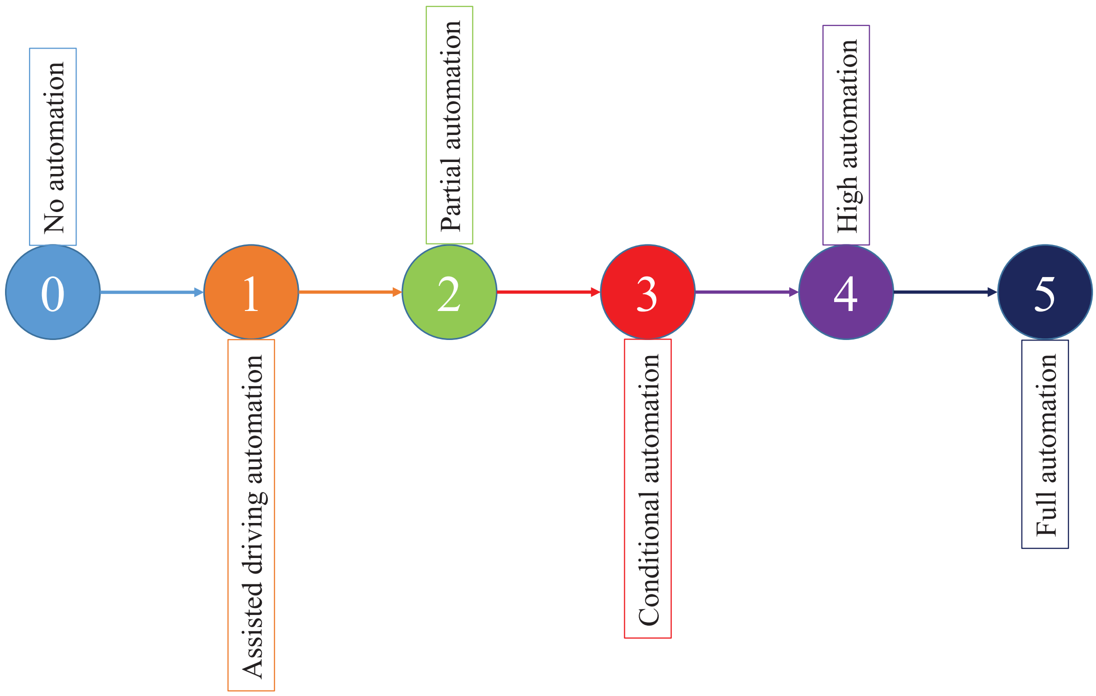


Figure 2.1: Different levels of automated driving defined by SAE [3].

Level 2 involves partial automation and allows vehicles to drive automatically under certain conditions, for instance vehicles can change lane to overtake in highways. However, the driver's presence is required at all times. Similarly, level 3 requires the presence of the driver although it is more advanced than level 2 in terms of making decisions using the vehicle's sensors (e.g., LiDAR). Level 4 and 5 bring about the automated vehicles where self-driving becomes possible. However, in level 4 when the condition of the road is difficult for driving, for example, there is visibility issues, the presence of the driver is needed [3].

The hardware components and the software algorithms of a CAV which are considered in this study are illustrated in Fig. 2.2. Fig. 2.2 shows that the information of surrounding environment of a CAV is perceived by the on-board sensors such as cameras, Lidars and radars as well as through V2V and V2I platforms. These information are fused within the perception block to create a map of surrounding and to localise the vehicle in the generated map [16]. The obtained vehicle pose and the map of the environment are then transmitted to other members of the traffic and to the path planning block, within the CAV. The path planning block which is the focus of this study generates a collision-free trajectory by calculating the optimal control actions with respect to vehicle dynamics and safety constraints. The calculated control actions are based on achieving some criteria that helps to minimise the energy consumption and travelling time. Finally, the control actions are realised by the vehicle's actuators.

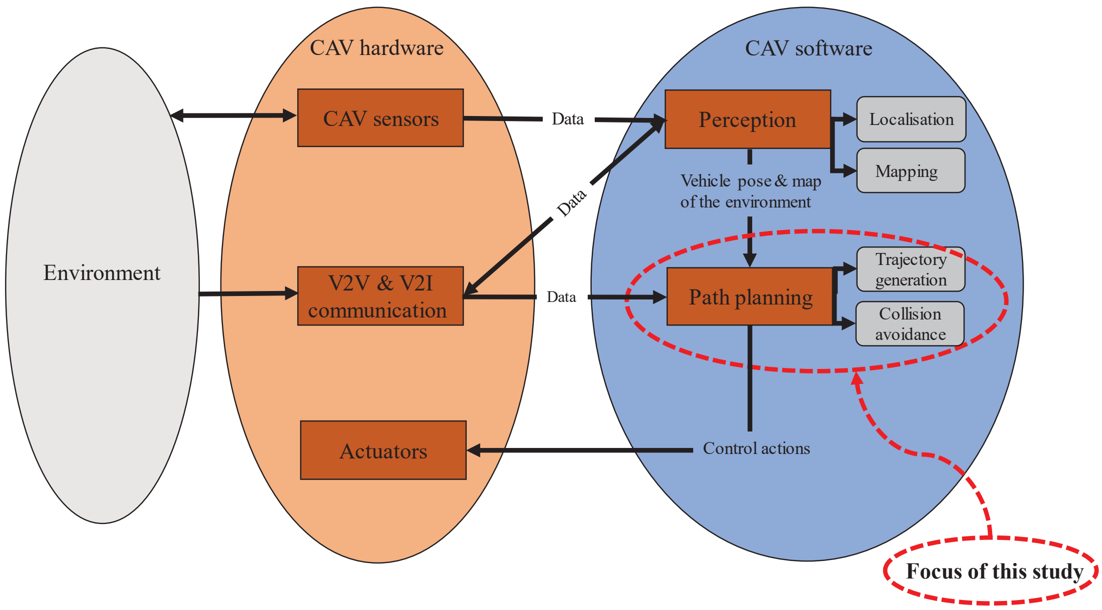


Figure 2.2: The software algorithms and hardware components of a CAV in this study.

2.1.1 Vehicle dynamics and modelling

A high-fidelity model of the vehicle dynamics and kinematics is a crucial requirement for analysing the behaviour and development of the model-based controllers of CAVs. Various vehicle models are employed for the planning task based on the trade-off between the model complexity and prediction accuracy. Several studies [17–19] modelled the behaviour of vehicles using second order dynamics and assumed the vehicle as a point-mass. This type of vehicle modelling does not consider the lateral behaviour of vehicles and hence the steering angle is not considered and the only control input is acceleration/deceleration. Other studies [20–23] employed a more sophisticated model of a vehicle, namely bicycle model, which also takes into account the lateral behaviour of vehicles. The bicycle model lumps the two front wheels and the two rear wheels in to a single wheel each to simplify the four wheel car into a two wheel bicycle. This study also represents the lateral behaviour of CAVs with the bicycle model [24]. The bicycle model of each CAV$_i$, $i \in \{1..N_v\}$ where N_v is the number of vehicles, consists of two degrees of freedom (DoF) which are sideslip angle β_i and yaw rate r_i, as in Fig. 2.3. The model also includes an additional DoF for the longitudinal velocity V_i. The equations of these DoFs along with other three to model the ground-fixed location, construct a set of differential equations to represent CAV$_i$ as follows:

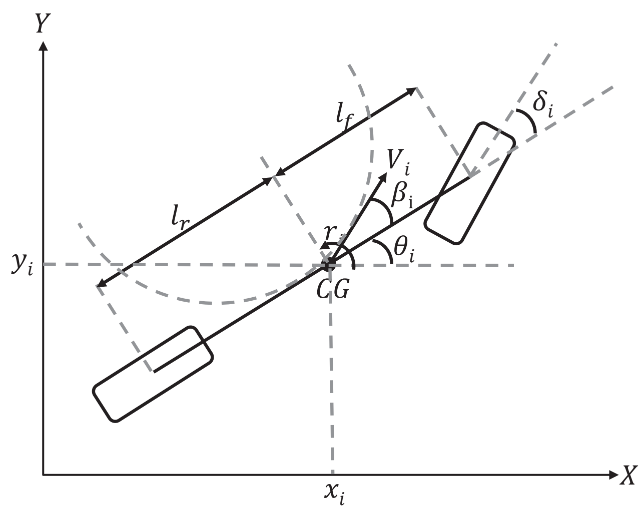


Figure 2.3: The bicycle model of CAV$_i$, i.e, $i \in \{1..N_v\}$.

$$\frac{d}{dt}\begin{bmatrix} r_i \\ \beta_i \\ V_i \\ x_i \\ y_i \\ \theta_i \end{bmatrix}(t) = \begin{bmatrix} \frac{\tilde{N_r}}{I_z \cdot V_i(t)} \cdot r_i(t) + \frac{N_\beta}{I_z} \cdot \beta_i(t) \\ (\frac{\tilde{Y_r}}{m \cdot V_i(t)^2} - 1) \cdot r_i(t) + \frac{Y_\beta}{m \cdot V_i(t)} \cdot \beta_i(t) \\ 0 \\ V_i(t) \cdot cos\theta_i(t) \\ V_i(t) \cdot sin\theta_i(t) \\ r_i(t) \end{bmatrix} +$$

$$\begin{bmatrix} 0 & \frac{N_\delta}{I_z} \\ 0 & \frac{Y_\delta}{m \cdot V_i(t)} \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a_i \\ \delta_i \end{bmatrix}(t), t \in [t_0, t_f]. \tag{2.1}$$

where $[r_i, \beta_i, V_i, x_i, y_i, \theta_i]^T$ and $[a_i, \delta_i]^T$ are, respectively, the system states and control inputs of CAV$_i$. $\mathbf{z}_i = [x_i, y_i, \theta_i]^T$ refers to the pose of CAV$_i$ in non-inertial reference system. $a_i(t)$ and $\delta_i(t)$ are, respectively, the acceleration (m/s^2) and steering angle (rad) of the vehicle. The constants m and I_z denote mass (kg) and moment of inertia $(kg.m^2)$ of the vehicle. t_0 and t_f represent the starting and final time (s) that a CAV is under control. In (2.1), it is assumed that the value of sideslip angle β_i is much less than one radian. The vehicle parameters $\tilde{N_r}$, N_β, N_δ, $\tilde{Y_r}$, Y_β and Y_δ are calculated as follows [24]:

$$\tilde{N_r} = l_f^2 \cdot C_F + l_r^2 \cdot C_R, \tag{2.2a}$$

$$N_\beta = l_f \cdot C_F - l_r \cdot C_R, \tag{2.2b}$$

$$N_\delta = -l_f \cdot C_F, \tag{2.2c}$$

$$\tilde{Y_r} = l_f \cdot C_F - l_r \cdot C_R, \tag{2.2d}$$

$$Y_\beta = C_F + C_R, \tag{2.2e}$$

$$Y_\delta = -C_F. \tag{2.2f}$$

where C_F and C_R are, respectively, the cornering stiffness of the front and rear tyres. l_f and l_r are distance of the front and rear axis from center of gravity of the vehicle. The vehicle parameters $\tilde{N_r}$ and $\tilde{Y_r}$ are in terms of the longitudinal velocity V_i which is omitted from (2.2a) and (2.2d) and it is directly imposed in (2.3).

To ensure CAVs drive within their dynamic limitations, the following constraints are enforced for each CAV_i:

$$\underline{V} \leq V_i(t) \leq \bar{V}, \tag{2.3a}$$

$$\underline{a} \leq |a_i(t)| \leq \bar{a}, \tag{2.3b}$$

$$\underline{\delta} \leq |\delta_i(t)| \leq \bar{\delta}, \tag{2.3c}$$

$$\underline{r} \leq |r_i(t)| \leq \bar{r}, \tag{2.3d}$$

$$\underline{\beta} \leq |\beta_i(t)| \leq \bar{\beta}. \tag{2.3e}$$

where $\bar{\cdot}$ and $\underline{\cdot}$ are, respectively, the upper and lower boundaries.

2.2 Path Planning Techniques for CAVs

Path planning is one of the fundamental functions of CAVs and there has been enormous research undertaken on this area [25–29]. Path planning is defined as moving a vehicle from a given point to a destination while avoiding obstacles and considering minimal energy consumption and travelling time as well as passenger comfort. Although path planning of CAVs for intersections is more challenging than areas such as highways, the technique of planning can be similar. This section reviews various techniques used in this area for both CAVs and robots. A summary of the path planning techniques is shown in Table 2.1.

2.2.1 Graph search based planners

Graph search methods are used for path planning due to their ability of guaranteeing optimality and simplicity of implementation. In this regard, *Dijkstra's algorithm* is a popular graph search method for finding the shortest path between any two nodes within a graph and is applied for path planning of CAVs in several studies [30, 31]. However, this algorithm performs a blind search which means every node is searched equally resulting in waste of time to search several unwanted nodes, particularly when the area of search is considerably large. *A* algorithm* which is an extended version of Dijkstra's algorithm improves the search computational time by augmenting the cost function with a heuristic function. This heuristic cost function of the A* algorithm helps to search towards the direction of the goal and avoids searching the unwanted nodes. A* algorithm is frequently used for path planning in its original version [32, 33] and in its improved versions namely dynamic A* [34], Theta* [35] and hybrid A* [36]. *State lattice* is also a graph search method that uses a discrete representation of the area to generate a gird [37]. The grid is then searched based on the local feasible actions of a vehicle to find the path. State lattice is a good planning algorithm in terms of handing several dimensions, however, the generated paths using this technique are shown to have discontinuity in curvatures and hence vehicles cannot perform evasive maneuvers [27–29]. Overall, graph search based methods can obtain optimal path, however, they require high resolution of the map and can be time consuming where the search area is large [38].

2.2.2 Optimisation based planners

Optimisation-based methods become popular to plan vehicles' path due to the recent developments in fast solvers of large optimisation problems, such as NPSOL [39] and IPOPT [40]. The main goal here is to formulate the planning as an optimisation problem and particularly as an OCP (see (2.4)) consisting of a desired objective function and the required constraints, which is solved using the available fast solvers [25, 26]. Prior research presented path planning problem of CAVs as an OCP with various objectives.

The studies in [25,41] set the objective function of the OCP to minimise a combination of the acceleration, its derivative and the error between the current speed and the speed limit. A similar OCP formulation is presented in [42] where the objective function is set to minimise the travelling time and the motion sickness of CAVs. Although these studies generated a path and achieved their objectives, solving their OCPs can be challenging, particularly when the problem involves multiple CAVs.

An OCP must be solved over an infinite horizon to generate an open-loop control strategies. However, solving such an infinite horizon OCP can be computationally expensive if not impossible. Model Predictive Controllers (MPCs), on the other hand, approximates the solution of an OCP over a finite horizon in a receding horizon fashion which can reduce the computational time [23, 43]. In addition, MPC can take into account the uncertainties of the dynamic environment as it periodically receives a feedback from the surrounding [44]. P. Falcone et.al [43] presented a simple hierarchical framework based on MPC for path planning and path tracking of CAVs. This work proposes one MPC for the high level controller to plan the vehicle's path based on a simplified point-mass vehicle model, and another MPC in the low level controller to track the generated path based on the bicycle model. The high level controller generates a path for the vehicles to reach the destination as fast as possible and the low level controller tracks the generated path based on active front steering. A similar work is presented in [45] where an MPC strategy is designed as a low level control which considers the coupled lateral and longitudinal forces of tyres.

Furthermore, the authors in [39, 46, 47] presented an MPC-based algorithm that employs a curvilinear coordinate system for tracking the roads. The curvilinear coordinate system tracks the centreline of the roads using the curvature as a function of the length of road's centreline [42]. The aim of the controller in the mentioned studies [39, 46, 47] is to generate the required control inputs to drive the vehicle in the centre of the road while minimising the energy consumption and subject to the vehicle dynamics. The paths obtained in these reports only consider the obstacle avoidance of static objects and ignore moving obstacles. The obstacle avoidance of dynamic objects such as cars is the major difficulty of path planning for CAVs.

2.2.3 Artificial potential field-based planners

Artificial potential field is also employed for path planning of CAVs due to its simplicity. This technique generates a collision-free path by considering the obstacles as the source of repelling forces and the target as the source of attractive force. The studies carried out in [22, 48] designed a path planning algorithm based on artificial potential field and MPC. These studies set the cost function of the MPC as potential functions of the road and obstacles. The controller inherits from the simplicity of potential field method and incorporates the vehicular and environmental dynamics within the MPC formulation.

Huang et al. proposed a similar work in [47] where a hierarchical algorithm for path planning is designed for vehicles' motion on highways. In the high-level controller, a resistance network is utilised to plan a lane change or a lane keeping behaviour. As the low-level controller, two MPC controllers are proposed for the lateral and longitudinal motions of the vehicle. Finally, as the longitudinal and lateral planning is performed separately a super twisting sliding mode control is employed to check the dynamic-feasibility and to track the generated path.

The major limitation of the studies in [22, 47, 48] is that their proposed algorithms are only tested for a maximum of fives vehicles. Moreover, the performance of algorithms in [22, 47] are only evaluated for highways which has less number of conflicts than urban roads and intersections.

2.2.4 Reinforcement learning based planners

Reinforcement Learning (RL) is also applied for path planning of CAVs. The goal of RL is to learn a policy through experience that maximises the total cumulative reward. In this regard, there exists two types of techniques that implement RL. The *model-based* technique learns the policy using a model of the system where as the *model-free* technique learns the policy without a model of the system. Both techniques are implemented using two types of algorithms called policy-based and value-based algorithms. The policy-based algorithms directly learn a policy that gain maximum future reward while value-based algorithms estimate the policy by estimating the associated value function.

The authors in [49] developed a value-based RL algorithm for the path planning of CAVs at signalised intersections. The proposed controller considers a positive or a negative reward value for different behaviours of the CAV. For example, a reward of 1 is considered to increases the speed for a faster travel and a reward of -1 is considered for when the time gap between two CAVs is less than 0.8 s to avoid collisions. A comparison of the proposed algorithm against HVs crossing the same signalised intersection shows that their algorithm drives the CAVs through the intersection with a faster average speed. Another value-based RL algorithm is developed in [50] with a focus on decentralised path planning of robots in a known environment. The controller of each robot optimises a value function which includes a reward term to keep a safe distance from other robots as well as to reach to the goal faster. It is shown that the proposed decentralised algorithm can achieve a collision-free path for up to 8 robots in real-time.

One of the major challenges of path planning using RL is learning within the shortest time which is a matter of how exploration of uncharted territory and exploitation of current knowledge are balanced [51]. In this light, there exists several algorithms that can keep a balance between such exploration and exploitation and help to avoid getting stuck in local optimum solutions [52]. Epsilon-greedy is among those algorithms that can keep a balance by choosing a probability of $\epsilon \in [0, 1]$ to explore or exploit more often [53].

2.3 Crossing of CAVs through Intersections

Intersections are one of the major area of collisions due to the high number of conflicts between vehicles travelling to different directions. In United States 2010, 47% of all accidents occurred at intersections and 28% of them lead to fatal injuries [15]. CAVs along with the technologies involved can provide improvements to these figures. Furthermore, CAVs can

Table 2.1: Summary of the path planning techniques for CAVs.

Technique group	**Algorithm**	**Algorithm description**	**Implemented in**
Graph search based planners	Dijkstra's algorithm	Searching from a specified source to all possible goals	[30, 31]
	A* algorithm family	Heuristic search from a specified source to a directed-goal	[32–36]
	State lattice	Decomposition of the environment in a local variable grid	[37]
Optimisation based planners	OCP	Solving an objective function along with constraints for the whole prediction horizon	[25, 25, 26, 41]
	MPC	Solving an OCP in a receding horizon fashion	[23, 43–45] [20, 39, 46, 47, 54]
Artificial potential field planners	Potential functions	An artificial force is calculated based on the repelling force of obstacles and attractive force of the target	[22, 47, 48]
Reinforcement learning based planners	Learning mechanism	A policy function generates actions based on the feedback from the environment	[49, 50, 55]

share with each other their location, speed and arriving time to the intersection which help them to cease the stop-and-go behaviour of current vehicles passing through intersections. This results in achieving better fuel consumption, emission and traffic throughput [56]. However, the collaboration between CAVs to optimally pass through intersections and the resulting improvement in traffic capacity are not fully understood yet, and this section provides a review of the state-of-the-art achievements in these areas.

2.3.1 Challenges

CAVs will be able to execute more complex manoeuvres than human drivers to cross intersections in a faster and safer manner with less energy consumption and higher traffic throughput [56]. However, these potential improvements require addressing three main challenges: collision avoidance, finding the minimum-time optimal solution instead of only checking the feasibility, and real-time implementation. Table 2.2 provides a summary of the challenges and the corresponding techniques.

Previous studies proposed three approaches to ensure collision avoidance among CAVs:

i) Reserving the whole intersection for one of the CAVs at a time; The authors in [57] formulated the intersection crossing problem of CAVs as an OCP with collision avoidance constraints that enforce CAVs to reserve the whole intersection for a period of time. The proposed OCP jointly improves energy consumption and passenger comfort. A similar work is presented by Tallapragada et al. in [17], where CAVs are split into clusters and each cluster reserves the whole area of the intersection for some time. The authors in [58, 59] introduced a scheduling method where CAVs are placed into a virtual lane based on their distance to the centre of the intersection and their risk of collision. Then, crossing time of the intersection is scheduled between the CAVs in the virtual lane. The study in [60] designs a similar intersection-reservation based crossing algorithm for CAVs and quantifies the energy consumption improvement of their algorithm as compared to HVs crossing the same intersection. It is shown that the proposed algorithm reduces the mean fuel consumption of every vehicle by 13.29–73.11% as compared to HVs crossing the intersection [60].

ii) Reserving a finite number of specific points (called conflict points) instead of the whole of the intersection; Mirheli et al. [61, 62] designed 16 conflict points for a four-leg intersection. Each leg of the intersection includes exclusively left turn and straight lanes. The proposed algorithm enforces CAVs to reserve the approaching conflict point(s) prior to their arrival. A more recent study of the conflict-point-reservation technique is presented in [63] where CAVs are capable of performing turning maneuvers. This study divides the intersection crossing problem of CAVs into two parts of firstly finding the passing sequence including the arriving time of CAVs to the intersection and then calculating the required control inputs to satisfy the obtained arriving time and passing sequence. The later problem is formulated as an OCP to minimise the energy consumption of CAVs and the former is also formulated and an optimisation problem which is solved using different search algorithms to compare their performance. It is shown that the best performance in terms of fuel consumption and travelling time is achieved when the passing sequence of CAVs is solved using the Monte Carlo Tree Search [63].

Another conflict point reservation approach is presented in [64] which allows flexible lane direction (e.g., incoming vehicles can travel to any outgoing lanes). In this work, a formation reconfiguration method is utilised to control longitudinal and lateral position of vehicles while avoiding collisions by reserving the conflict points. It is shown that crossing intersections with flexible lane direction technique outperforms signalised intersections and unsignalised intersections with fixed lane direction in terms of traffic throughput. A similar work is provided in [65] that designs an intersection crossing algorithm with a focus on erasing lane changes. This work also avoids collisions by reserving the conflict points through solving an optimisation problem that yields the optimal collision-free arriving times to the conflict points.

iii) Utilising freely the whole space of intersections, a.k.a. lane-free crossing; Generally speaking, reservation-based collision avoidance approaches require CAVs following predefined paths that do not allow the vehicles to fully exploit the intersection area. Therefore, reservation-based collision avoidance approaches are not efficient in terms of reducing travelling time and energy consumption. Prior studies developed lane-free intersection crossing algorithms based on OCP in [2, 66]. To avoid collisions, the Euclidean distance between any pair of CAVs are constrained to be greater than a safe margin. This formulation of the collision avoidance constraints is non-convex and non-differentiable [67], and hence any optimisation problem including them are difficult to solve. Li et al. in [66] divided the non-convex problem of intersection crossing into two stages to make it tractable. At stage one, CAVs inform the central controller with their intention and then make a standard formation which is computed online. At stage two, the controller searches an offline constructed lookup table for the intended crossing scenario and finds the control inputs of each CAV. The authors suggested to solve offline an individual optimal control problem for any possible crossing scenario to construct the lookup table of the control commands. However, the resulting offline problems are still non-convex and solving them for all possible scenarios of 24 CAVs take around 358 years [66]. Alternatively, Li et al. in [2] fixed the crossing time to a constant value and converted the minimum-time optimal control problem to a feasibility problem that is solved online.

As the second challenge, the above-mentioned algorithms that only find a feasible (collision-free) solution to the problem of the intersection crossing do not fully exploit the CAVs' advantages to minimise the crossing time. In other words, minimising the crossing time is not part of their objectives. The studies carried out in [17] and [2] focus on the passenger comfort and address this challenge by minimising fluctuations of the vehicles' acceleration. Other researchers in [68–71] optimised the motion of CAVs to move on the predefined paths, reserving conflict points, with as close speed as possible to the limit of the intersection rather than directly finding the minimum-time paths. However, in complex scenarios, these paths need to be obtained as the solution of a minimum-time OCP instead of heuristically. The authors in [66] formulate the intersection crossing problem of CAVs as a minimum-time OCP to minimise the crossing time without any restrictions on the crossing paths (except the road boundaries). However, their algorithm is non-differentiable and difficult solve.

Finally, it is always challenging to implement optimal control strategies in real-time. CAVs are intelligent agents communicating to each other and to the infrastructure to share information for collaborating and achieving global objectives. The optimal strategies for crossing intersections, therefore, must operate on a network of cars (i.e., networked controller), which are seeking their own conflicting objectives (like minimising their individual crossing times). A centralised topology with a fully available information of all the CAVs or different decentralised topologies, where the CAVs are fully or partially connected, can be used to calculate the optimal crossing strategy of all the CAVs. The centralised controllers receive information of all vehicles, compute trajectories, and send back the calculated trajectory of each individual CAV. There is no path planning at the CAV level and vehicles only follow the provided trajectories. Li et al. [2, 66] proposed a centralised, but computationally expensive, optimal controller for multiple CAVs crossing a lane-free intersection. The centralised algorithms in [69, 70] split the problem into two stages, finding the crossing order and calculating the control inputs to follow the attained crossing orders, to make it computationally tractable, however, at the cost of obtaining sub-optimal solutions.

In the decentralised strategies, on the other hand, each CAV computes its own trajectory that also approximates a solution to achieve a level a global objective. Although decentralised algorithms find sub-optimal solutions, it is shown to be less computationally expensive as compared to the centralised counterparts [18, 72].

The authors in [18, 72] formulated a decentralised OCP controller for CAVs crossing intersections where each CAV has access to the shared information of all the others. However, CAVs which are crossing an intersection cannot be practically fully connected to each other at all the times. This means that at any instance of time, each CAV only communicates with a subset of the others, i.e. partially connected. Bian et al. [73] proposed a framework where the CAVs travelling on the same lane can communicate to each other, while they estimate the states of the other not-connected vehicles. Reference [59] proposes a partially connected distributed algorithm based on the concept of virtual platooning. CAVs, first, form a virtual platoon and then optimise their arriving time to the intersection to avoid collision. This is a decentralised reservation-based algorithm that allows only one CAV at a time to be within the intersection. Generally speaking, unlike the centralised controllers which are capable of finding the global optimum solution, decentralised controllers can only find sub-optimal strategies [74].

Table 2.2: Summary of the challenges of intersection crossing and the corresponding techniques.

Challenge	**Techniques to address**	**Implemented in**
Collision avoidance	Reservation of the whole Intersection	[17, 57–60, 75]
	Reservation of conflict points	[1, 61, 62]
	Lane-free	[2, 66]
Finding the minimum-time optimal solution instead of only checking the feasibility	Minimisation of fluctuation of the vehicles' acceleration	[2, 17]
	Minimisation of deviation from the speed limit	[68–71]
	Minimisation of the crossing time	[66]
Real-time implementation	Centralised strategies with fully-observable data	[2, 66, 69, 70]
	Decentralised strategies with fully connected CAVs	[18, 72]
	Decentralised strategies with partially connected CAVs	[59, 73]

2.3.2 Traffic capacity of intersections

Capacity analysis of intersections is essential for traffic management and for the planning of transport systems. For instance, such analysis for signalised intersections can help traffic engineers to identify the required green time for each phase of a traffic light to achieve a better traffic throughput. Moreover, capacity analysis of intersections when CAVs are crossing can help to compare various intersection crossing algorithms and recognise the algorithm that achieves the highest traffic throughput. However, the capacity measurement of intersections is different depending on whether CAVs or HVs passing through. CAVs can communicate with each other and with the infrastructure [76], resulting in less response and reaction time than HVs [77]. In addition, CAVs' collaboration make them able to keep a much shorter headway h_d (i.e., the time difference between the front of the lead vehicle passing a point and the front of the following vehicle passing the same point as in Fig. 2.4) from a preceding vehicle. There are extensive prior works to char-

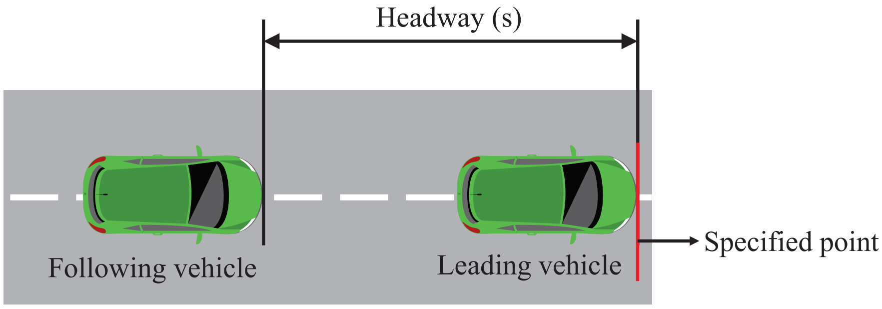


Figure 2.4: A graphical representation of headway.

acterise the capacity of intersections for HVs (e.g. [78–80]), however, such analysis is still an open research topic for CAVs. The remaining of this section explains the measures of intersection capacity firstly for HVs and then for CAVs.

Capacity of intersections with HVs

The measures of capacity for HVs crossing both signalised and unsignalised (two-way stop-controlled (TWSC) and all-way stop-controlled (AWSC)) intersections are extensively discussed in Highway Capacity Manual [78]. The manual introduces a measure to quantify the capacity of the unsignalised TWSC and AWSC intersections based on, respectively, gap acceptance and queuing theories. According to the gap acceptance theory, the capacity of TWSC intersections are modelled based on the distribution of major stream gaps, driver's response time in choosing those gaps and follow-up time [81]. Two important parameters namely *critical gap time* t_c (s) and *follow-up time* t_f (s) need to be defined which are both obtained through traffic data in real world scenarios. The critical gap time is defined as the minimum time that a vehicle travelling in a minor-street can accept to fill a gap in the major-street, and the follow-up time is described as the time between the departure of one vehicle from a minor-street and the next vehicle from the same minor street. The potential capacity of unsignalised TWSC intersections which is the potential capacity of minor movement x is defined in [78] as follows:

$$C_{p,x} = V_{c,x} \frac{e^{-V_{c,x} t_{c,x}/3600}}{1 - e^{-V_{c,x} t_{f,x}/3600}} \tag{2.4}$$

where $C_{p,x}$ is the potential capacity of minor movement x (veh/h) and $V_{c,x}$ is the flow rate for movement x (veh/h).

Furthermore, the capacity of unsignalised AWSC intersections is modelled based on the queuing theory by calculating the headway h_f of HVs for each lane [78]. The headway of HVs is calculated in an iterative approach due to the interdependence of the traffic flow on all approaches of the intersection (a detailed procedure of calculating the capacity of AWSC intersections is provided in Chapter 4.1). Meanwhile, it is recommended in [78] to calculate the capacity of signalised intersections as the saturation flow rate times the green time ratio. The saturation flow rate is defined as the maximum number of vehicles, queued in an approach, that can pass through the intersection in an hour (equivalently $\frac{3600}{h_f}$) [78]. The green time ratio is the effective time of a traffic light's green phase as compared to the whole cycle length.

All of the above-mentioned measures assume that saturated headway (e.g, the constant headway achieved once a stable moving queue is established) of HVs in the queue of lanes is around 1.9 s which is obtained through traffic data in the USA and it is referred to indicative HCM headway in this study. This assumption makes these measures inappropriate for CAVs crossing intersections where the headway of CAVs is much smaller and almost the same for all the vehicles in the queue. In fact, the headway of CAVs depends on the controller design of the CAVs. Previous studies employed a wide range of headway values for CAVs. For instance the authors in [82,83], designed their CAV controllers based on a fixed headway of 0.9 s. Other studies [84–86] employed a stochastic headway with a value from 0.5 s up to 2 s based on four modes namely aggressive, neutral, conservative, and safe. Therefore, it can be observed that capacity measurement of intersections when CAVs passing through them can be different for each controller design and the chosen headway value.

Capacity of intersections with CAVs

Whilst human reaction is the dominant factor to measure capacity of intersections with HVs, reaction time is not a dominant factor of the capacity of intersections when CAVs cross in a lane-free order. This is because of significantly shorter reaction time of CAVs as compared to HVs. In addition, CAVs' collaboration make them able to keep a shorter safety distance from each other than HVs. Therefore, the previous measures of intersection capacity for HVs are not applicable for CAVs.

To evaluate capacity of the intersections with CAVs, the authors in [77,87,88] employed the same measure that is defined in HCM [78] for the unsignalised intersections, though with a new headway definition for CAVs. In [87], intersections are assumed as service providers and CAVs' headway is redefined as service time (i.e., crossing time) which is derived by applying queuing theory. The service time is based on the safety time gap of CAVs approaching the intersection from the non-conflicting and the conflicting streams.

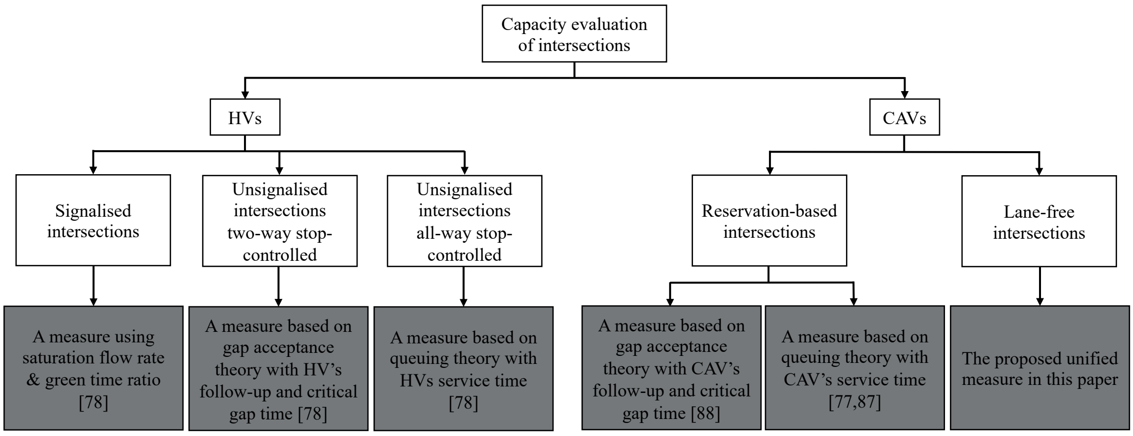


Figure 2.5: The measures for capacity evaluation of intersections (grey boxes) with both the human-driven vehicles (HVs) and connected and autonomous vehicles (CAVs).

The capacity of the intersection is then modelled as a function of the service time (i.e., $\frac{3600}{E(S)}$ where $E(S)$ is the service time in terms of the safety gaps) [87].

A similar work is proposed in [77] that employs the M/G/1 queue model to drive an equation for the capacity of the intersections. This model assumes that the intersection capacity is equivalent to the service rate of vehicles. The intersection capacity equation is then derived in terms of the probability of vehicles arriving from conflicting stream and non-conflicting stream with different service rate for each stream. Finally, the authors in [88] reformulated the capacity measure of the unsignalised TWSC intersections to use the critical gap and follow-up time of CAVs instead of the ones of HVs. The measures provided by these researchers are effective to evaluate capacity of the intersections when CAVs drive through a restricted set of lanes, however, are not applicable to the lane-free intersections. This is majorly because headway and service time are both defined for when vehicles drive within lanes, however, in a lane-free traffic flow vehicles do not have such a restriction. Hence, there is a need for a measure to quantify the capacity for the lane-free crossing of CAVs through intersections.

Fig. 2.5 summarises different measures that are proposed by prior works to calculate the capacity of intersections for both HVs and CAVs.

2.3.3 Lane-free crossing of CAVs through intersections

Fig. 2.6 illustrates an example layout of a lane-free and signal-free intersection. The figure includes three CAVs which are moving from their initial points, depicted with the most solid colour, towards their intended destinations which are with the most transparent colour. The intersection comprises of four approaches, each of them has a separate incoming and outgoing lane. In a lane-free intersection, vehicles can freely change their

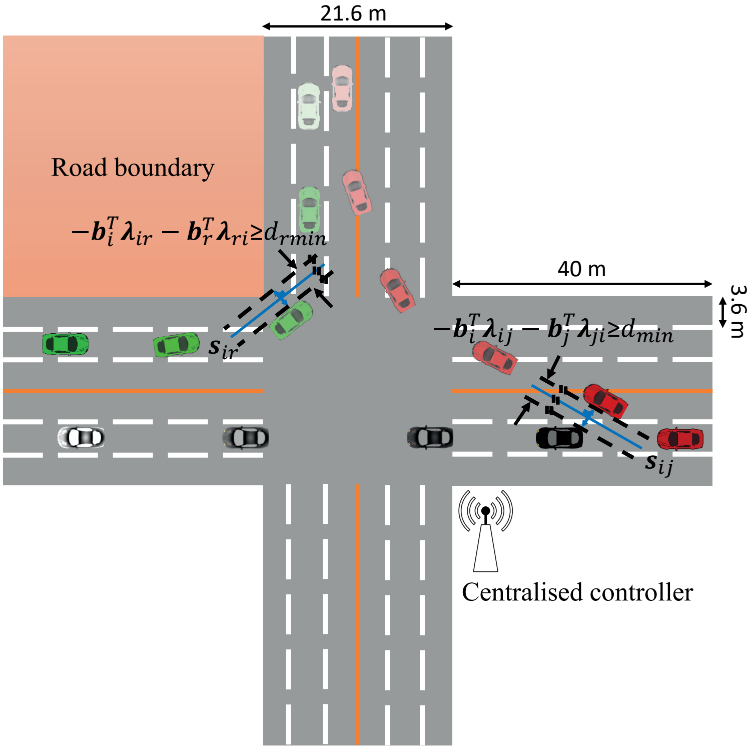


Figure 2.6: Layout of the studied lane-free and signal-free intersection which also shows the sufficient conditions for obstacle avoidance. Further details are presented in section 3

lanes in favour of faster crossing through the intersection. For instance, Fig. 2.6 shows the red CAV overtakes the black CAV by using the opposite lane.

In this study, the intersection does not have traffic lights because the CAVs can directly communicate their states and intentions. There is a coordinator that receives all the information from the CAVs when they arrive to the control zone and centrally control them to efficiently and safely cross the intersection. The control zone is defined based on the communication range of the coordinator, that is assumed 50 m. The coordinator then counts the number of vehicles entering the zone and when the number of vehicles reaches its practical limit or one of the vehicles reaches the beginning of the intersection, the coordinator calculates a safe trajectory for all the vehicles within its range. There is no human-driven vehicle or pedestrian. To compare the results with the ones of the prior research, this paper assumes that CAVs drive within the lanes before and after the control zone, which is shown to have no effects on the provided results (the results depend on the longest path of travelling CAVs). The authors in [89, 90], on the other hand, suggest that CAVs will drive in a seamlessly lane-free order within and outside intersections. In other words, there will not be a separate controller for different sections of roads and CAVs continuously collaborate to reach their final destinations without collision.

As explained in Sections 3.2 and 3.3, the expression $-\mathbf{b}_i^\top \boldsymbol{\lambda}_{ij} - \mathbf{b}_j^\top \boldsymbol{\lambda}_{ji} \geq d_{dmin}$ in Fig.

2.6 is the dual representation of the distance between a pair of CAVs, and $\boldsymbol{s}_{ij}$ is the separating hyperplane placed between them. The parameters $\mathbf{b}_i$ and $\mathbf{b}_j$ are related to the size of CAV$_i$ and CAV$_j$ respectively and $\boldsymbol{\lambda}_{ij}$, $\boldsymbol{\lambda}_{ji}$ and $\mathbf{s}_{ij}$ are dual variables. Similarly, $-\mathbf{b}_i^\top \boldsymbol{\lambda}_{ir} - \mathbf{b}_r^\top \boldsymbol{\lambda}_{ri} \geq d_{rmin}$ is the dual representation of distance between the red CAV and a road boundary and $\mathbf{s}_{ir}$ is the separating hyperplane between them. $\mathbf{b}_r$ is related to the size of the road boundary and $\boldsymbol{\lambda}_{ir}$, $\boldsymbol{\lambda}_{ri}$ and $\mathbf{s}_{ir}$ are the dual variables. For further details, the reader is referred to Sections 3.2 and 3.3.

2.4 Optimal Control Problem (OCP)

As previously mentioned in Sections 2.2.2 and 2.3.1, OCPs are frequently used to formulate and solve the path planning of CAVs as well as the lane-free crossing of CAVs through intersections. OCPs take into account the dynamic of a system in finding the optimal control law u^* that minimises a cost function, subject to other equality and inequality constraints. A general form of a continuous-time infinite horizon OCP with a cost junction of J subject to system dynamics f is presented in (2.5) [4].

$$u^*(.) = \underset{u(.)\in\mathcal{U}^\infty}{\arg\min} \quad J_\infty(x, u; t) := \int_t^\infty L(x(t), u(t))dt \tag{2.5a}$$

subject to:

$$\dot{x} = f(x(t), u(t)), \tag{2.5b}$$

$$h(x(t), u(t)) \leq 0, \tag{2.5c}$$

$$g(x(\infty)) = 0, \tag{2.5d}$$

$$x(t_0) = x_0, \tag{2.5e}$$

$$x(t) \in X, u(t) \in \mathcal{U}, \tag{2.5f}$$

$$\forall t \in [t, t + \infty].$$

where (2.5c) represents the inequality constraints of the control inputs u and the states x and (2.5f) presents the corresponding boxing constraints. The initial and final values of the states are specified in, respectively, (2.5e) and (2.5d) as boundary constraints. The cost function J includes a Lagrange term L that penalises the distance between the current and the reference trajectory for the whole period of prediction. To solve the OCP given in (2.5), different solution methods are explained in Section 2.6.

It is worth noting that OCPs are a class of dynamic optimisation problems because they have ordinary or partial differential equations (PDE) as constraints, making them different from static optimisation problems. Such dynamic constraints require solving the OCP for the infinite horizon (instead of a single sampling point) because any action at a sampling point will affect the whole future [91].

2.4.1 Model predictive control (MPC)

The major bottleneck of formulating path planning problems of CAVs as OCPs is that it must be then solved over the infinite horizon. It means that at any time t, the whole optimal trajectory from the t until the terminal time $t + \infty$ must be calculated. However, solving such an infinite horizon OCP, particularly with a large number of states, is computationally expensive if not impossible (except for special cases). MPC, on the other hands, solves the OCP in (2.5) over a finite horizon in a receding horizon fashion which can reduce the computational time [4]. Solving the OCP in such receding horizon fashion results in taking into account the disturbances as well as the mismatch behaviour of the model and real plant. MPC is a closed-loop controller that receives feedback from the plant after applying any control signal and updates the current states of the system before resolving OCP for the next receded horizon. This helps MPC to compensate the errors due to limiting the horizon. A general form of the OCP which is solved within the MPC framework is provided in (2.6) which is the discritised verison of (2.5) over a finite horizon N_p.

$$u^*(.) = \underset{u(.)\in\mathcal{U}^{N_p}}{\arg\min} \quad J_{N_p}(x, u; n) := \sum_{k=n}^{n+N_p-1} L(x(k), u(k)) + \mathcal{M}(x(n + N_p)), \tag{2.6a}$$

subject to:

$$\dot{x} = f(x(k), u(k)), \tag{2.6b}$$

$$h(x(k), u(k)) \leq 0, \tag{2.6c}$$

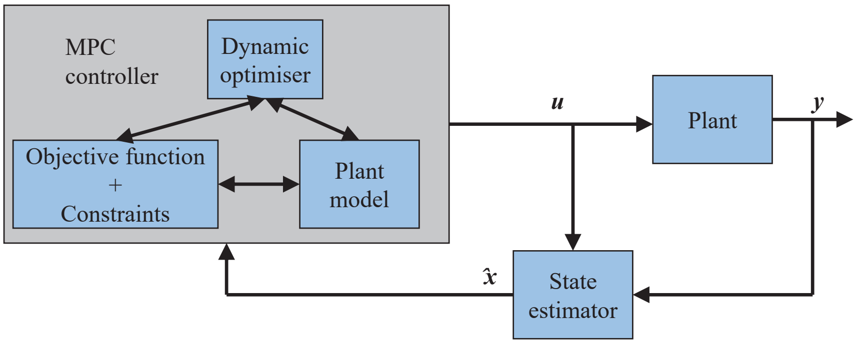
$$g(x(n + N)) = 0, \tag{2.6d}$$

$$x(n) = x_0, \tag{2.6e}$$

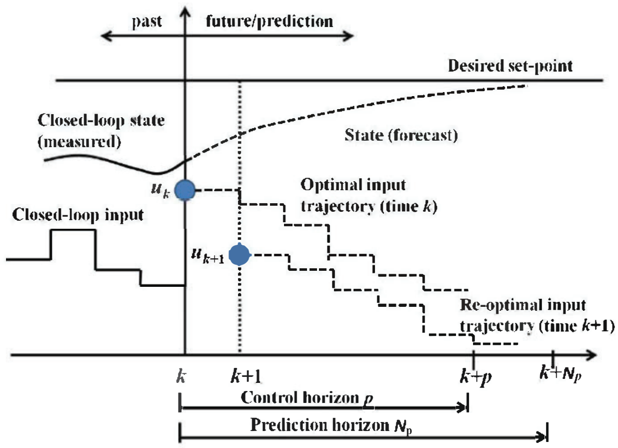
$$k \in [n, n + N_p - 1], \tag{2.6f}$$

$$x(k) \in X, u(k) \in U. \tag{2.6g}$$

The objective function (2.6a) includes a stage cost as the Lagrange term $L(.)$ and a Mayer term $\mathcal{M}(.)$. While the Largragne term is for the measurement of the running costs of the states and the control inputs during the prediction horizon, the Mayer term penalises any deviations of the states at the end of prediction horizon from their desired final values. Although it is shown that the existence of the Mayer term in the objective function can be effective for the stability and convergence of the system, it is not a necessary condition. There exists new techniques that can improve the stability of MPCs without adding terminal constraints (e.g., see [5, 92]).



(a)



(b)

Figure 2.7: Overview of (a) block diagram and (b) functionality of MPC (courtesy of [4]).

Figure 2.7 (a) shows a general block diagram of how MPC controls a plant. From figure 2.7 (a), it can be observed that an MPC controller consists of an objective function as well as constraints, a dynamic optimiser and a plant model. The MPC controller calculates the optimal control inputs $u^*(.)$ and sends them to the plant model. The current state values of the system are then either measured or estimated by a state estimator block to provide a feedback to the MPC controller as initial values.

Figure 2.7 (b) illustrates the functionality of an MPC controller. It can be seen that an optimal control input $u^*(.)$ is calculated for any sampling instant k for a finite prediction horizon $T = N_p \times h$ where h is the sampling length. The optimal control input is calculated by minimising the cost function of the controller which can be related to the required time for the states to reach the defined set points or the maximum level of overshoots. A feedback which is the current measurement of the system is used as the initial states to construct an OCP. The OCP is then solved and the first sequence of p, the calculated control input, is applied to the plant. The controller then updates the states using the new measurements at $(k + p)h$ and shifts the prediction horizon and control prediction forward to find the new optimal control inputs [4]. A summary of the operation of MPC controllers is as follows:

1. Solving an open-loop finite horizon OCP to obtain optimal control sequence $u^*(.)$.
2. Applying the first control input to the plant and measuring or estimating the output.
3. Using the output as a feedback to initialise and resolve the OCP while shifting the horizon one step forward.

It is worthy to note that solving the general form MPC in (2.6) for multi-agent systems such as path planning of multiple CAVs in a centralised frame can still be impractical. This is majorly because solving the MPC in a centralised framework assumes that all the information of the systems/CAVs (e.g., position, speed, ect.) are available, however, this is not the case in reality. As an example, CAVs approaching an intersection from different directions are not able to share information at all times due to the limitation of the communication range of each CAV. Moreover, controlling CAVs in a centralised frame can involve issues related to safety such that failure of the centralised system results in failure of all CAVs. CAVs are intelligent agents that can directly communicate with other members of the traffic without the need of a centralized controller. In this light, a distributed control strategy is desirable where each CAV finds its own control inputs by collaborating with others.

2.5 Optimisation Problems

The goal of an optimisation problem is to find the best solution out of all the feasible solutions which are defined by a set of equality and inequality constraints. A general form of an optimisation problem is as follows:

$$\text{minimise} \quad f(x) \tag{2.7}$$

$$\text{subject to:}$$

$$h_i(x) \leq 0, \quad i = 1, ..., m,$$

$$g_j(x) = 0, \quad j = 1, ..., p.$$

where the domain $\mathcal{D} = \left(\cap_{i=1}^{m} \mathbf{dom}_{h_i}\right) \cap \left(\cap_{j=1}^{p} \mathbf{dom}_{g_i}\right)$ is nonempty and the optimal value of (2.7) is d^* (e.g., $d^* = inf\{f(x) \mid \forall x;\ h_i(x) \leq 0,\ i = 1, ..., m,\ g_j(x) = 0,\ j = 1, ..., p\}$). The optimisation problem (2.7) is solved for x that minimises the objective function $f(x)$ and meets $g_j(x) = 0, j = 1, ..., p$ and $h_i(x) \leq 0, i = 1, ..., m$.

Before solving any optimisation problem it is important to check if it is convex. Solving non-convex problems is in general harder than convex problems due to the existence of several local minimums and saddle points. On the other hands, when solving a convex problem, the existence of a global optimal solution is always guaranteed [93]. This difference between a convex and non-convex function is depicted in Fig. 2.8.

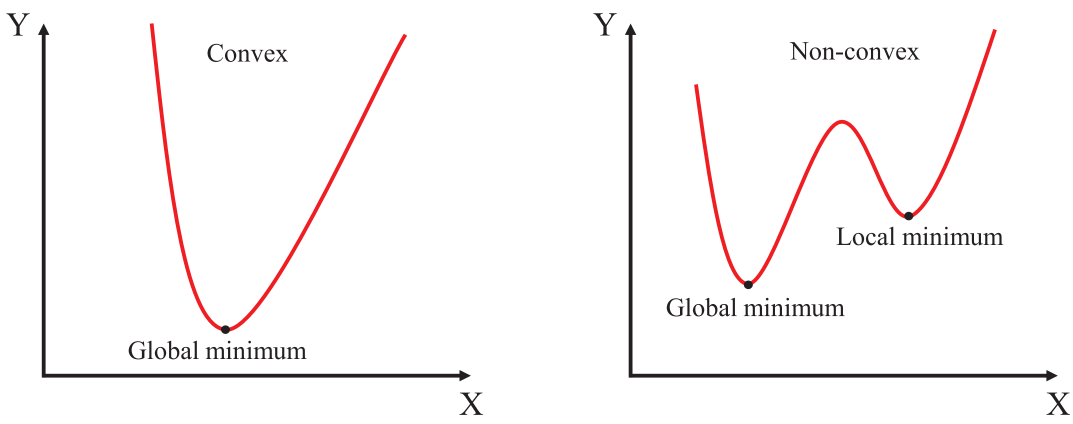


Figure 2.8: An example of a convex and non-convex functions.

In a formal way, a function f where $f : I \rightarrow \mathbb{R}$ and I is a nonempty interval of $\mathbb{R}$ is said to be convex if the following condition holds [94]:

$$f(\alpha x + (1 - \alpha)x\prime) \leq \alpha f(x) + (1 - \alpha)f(x\prime) \qquad \forall x, x\prime \in I, \quad \alpha \in [0, 1] \tag{2.8}$$

Geometrically, if a line connecting any two points of function f does not lie below its graph. This geometrical property of a convex function which also describes equation (2.8) is shown in Fig. 2.9.

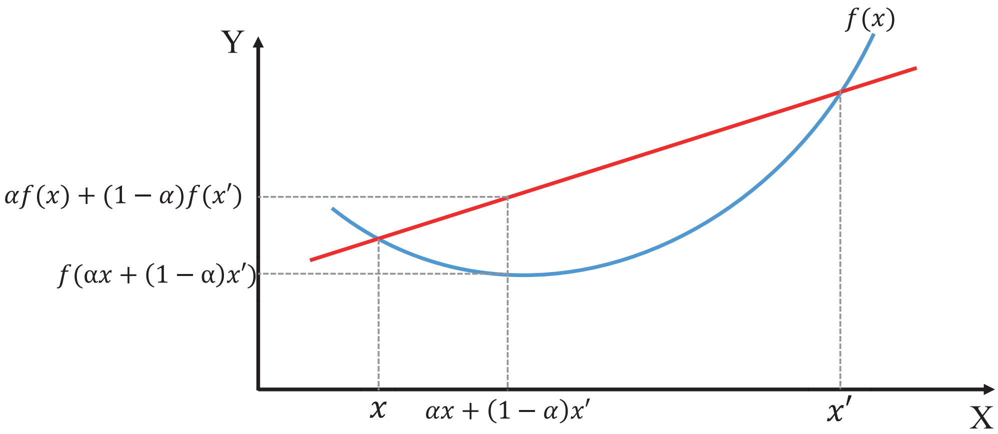


Figure 2.9: The fundamental property of a convex function.

To discuss convexity in optimisation problems consider the general optimisation problem given in (2.7). In order for problem (2.7) to be convex the following requirements must be satisfied [95]:

- the objective function $f(x)$ must be convex.
- the inequality constraints $h_i(x) \leq 0, \quad i = 1, ..., m$ must be convex.
- the equality constraints $g_j(x) = 0, \quad j = 1, ..., p$ must be affine, e.g. $g_j(x) = a_j^T x - b_j$.

Any optimisation problem that does not follow the above-mentioned conditions (e.g., has a non-convex cost function or a non-convex constraint) is considered as a non-convex problem and solving it can be computationally expensive.

2.5.1 Dual problem theory

As the optimisation problem (2.7), considered as primal, is a minimisation problem, its dual is a maximisation problem and the solution to the dual problem is a lower boundary on the optimal value of the primal problem [95]. In general, the solution of the primal and dual problems are not equal and the different between their solutions is known as the duality gap. However, if some *constraint qualifications* such as Slater's condition is satisfied, the duality gap is zero and the solution of both problems becomes equal [95]. Slater in [96] states that if the primal problem is convex and there exists a strictly feasible solution $\hat{x}$ to the primal problem, strong duality holds and the solution of both problems becomes equal.

To derive the dual problem of the primal problem (2.7) the Lagrangian function is employed and as a result the dual problem is also referred to as Lagrangian dual problem.

The Lagrangian function of the primal problem (2.7) is shown in (2.9) which considers the objective function with weighted sum of the constraint functions [95]:

$$\mathscr{L}(x, \lambda, \upsilon) = f(x) + \sum_{i=1}^{m} \lambda_i h_i(x) + \sum_{j=1}^{p} \upsilon_j g_j(x). \tag{2.9}$$

where λ_i and υ_j are called the Lagrange multiplier or dual variables associated with the ith inequality and jth equality constraints respectively.

The dual function $\mathcal{G} : \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$ where $\lambda \in \mathbf{R}^m, \upsilon \in \mathbf{R}^p$ is then formulated to have minimum value of Lagrangian over x as follows:

$$\mathcal{G}(\lambda, \upsilon) = \inf_{x \in \mathcal{D}} \mathscr{L}(x, \lambda, \upsilon) = \inf_{x \in \mathcal{D}} \left(f(x) + \sum_{i=1}^{m} \lambda_i h_i(x) + \sum_{j=1}^{p} \upsilon_j g_j(x) \right). \tag{2.10}$$

The dual function (2.10) leads to the following optimisation problem which seeks for the pair (λ, υ) while $\lambda \geq 0$:

$$\begin{aligned} &\text{maximise} \quad \mathcal{G}(\lambda, \upsilon) \\ &\text{subject to:} \quad \lambda \geq 0. \end{aligned} \tag{2.11}$$

As previously mentioned, the dual problem (2.11) yields a lower bound on the optimal value of the primal problem (2.7) (e.g., $\mathcal{G}(\lambda, \upsilon) \leq d^*$). However, if Slater's condition is satisfied, the solution of both problems are equal $\mathcal{G}(\lambda^*, \upsilon^*) = d^*$. Moreover, the dual problem (2.11) is a convex optimisation problem because the objective function which is to be maximised is concave and the constraint is convex. This is true regardless of whether the primal problem is convex or non-convex [95].

2.5.2 Differentiable functions

Differentiable functions enable the use of fast gradient-based solvers for calculating the optimal solutions of complex optimisation problems. A function is said to be differentiable if there exists a derivative for each point in its domain [97]. Function $f : U \to \mathbb{R}$ is differentiable at point $a \in U$ with $f'(a)$ if:

$$f'(a) = \lim_{h \to 0} \frac{f(a+h) - f(a)}{h} \tag{2.12}$$

A differentiable function is also continuous but the converse does not hold. This is simply because for a differentiable function the derivative should exist at all points in its domain. Therefore, the only way for the derivative to exist is that the function also exists which indicates continuity. The absolute function $f(x) = |x|$ as an example is a continuous function at all points in its domain but not a differentiable one at $x = 0$. Generally speaking, differentiable functions do not contain angle, break or cusp.

A multivariable function $f : \mathbb{R}^p \to \mathbb{R}^q$ is said to be differentiable at a point a if there exists a linear map $J : \mathbb{R}^p \to \mathbb{R}^q$ such that:

$$\lim_{h \to 0} \frac{||f(a+h) - f(a) - J(h)||_q}{||h||_q} = 0. \tag{2.13}$$

If function f is differentiable at point a, the linear map J is defined by the Jacobian matrix which includes all the partial derivatives.

2.6 Numerical Methods for Solving OCPs

Solving OCPs can be challenging due to several reasons including large dimension, as well as non-linearity, non-convexity and non-differentiability of the problem. Although convex optimisation problems with linear constraints can be solved efficiently using the state-of-the-art algorithms, optimisation problems that include nonlinear constraints are computationally expensive [95].

Previous researchers proposed three approaches to solve OCPs of the form (2.5) [98, 99]: i) dynamic programming which is based on the Bellman's principle of optimality; ii) indirect methods that are based on Pontryagin's minimum principle; and iii) direct methods that convert OCPs into NLPs and attempt at solving the NLPs using an NLP solver.

2.6.1 Dynamic programming (DP)

Dynamic programming (DP) finds the solution of an OCP by breaking it down into sub-problems. The sub-problems are then solved and their solutions are combined to obtain an overall solution. Such breakdown of the problem into sub-problems is performed based on the Bellman's principle of optimality. This principle states an optimal policy has the property that regardless of the first state and first sample of control sequence, the remaining control sequence construct an optimal control policy with regard to the state resulting from the first control sequence [100].

DP utilises the value function of OCPs to iteratively solve them. A value function yields the value of an objective function at a solution. The optimal value function of the infinite-horizon OCP in (2.5) with an initial state $x(t) = x_0$ is as follows:

$$V_{\infty}(t, x_0) = J_{\infty}(x^*(.), u^*(.); t, x_0) := \underset{u(.)\in U}{\text{minimise}} \quad J_{\infty}(x, u; t, x_0) \tag{2.14}$$

DP employs Bellman's equation, given in (2.15), to iteratively calculate an optimal value function of an OCP with the finite horizon N (i.e., OCP_N) from an initial value V_0 and with respect to the system dynamics $x(k+1) = f(x, u; k)$ and the Lagrange term $L(x, u; k)$ [101].

$$V_{K+1}(x; k) = \underset{u(.)\in U}{\text{minimize}} \ \{V_K(f(x, u; k)) + L(x, u; k)\} \tag{2.15}$$

where $K \in [1, N_p]$, $k \in [0, N_p - 1]$ and $V_0 = 0$.

Solving an OCP_{N_p} given as in (2.5) with DP leads to $N_p - 1$ overlapping sub-problems. These sub-problems can be formulated using (2.15) which is based on the value of the current problem and the remaining sub-problems. Moreover, such formulation and solving the sub-problems can be preformed with two techniques called *memoization* or *tabulation*. The memoization technique solves the sub-problems from top to bottom. The solution of sub-problems are saved in a lookup table and are used to obtain an overall solution recursively. On the other hand, the tabulation technique solves the sub-problems from bottom to top and the overall solution is built up using previously calculated solutions, hence no recursion.

The Bellman's equation in (2.15) is in discrete-time form and the corresponding continuous-time is Hamilton-Jacobi-Bellman (HJB) equation. HJB converts the OCP into a set of Partial Differential Equations (PDEs) with boundary conditions (e.g., initial and final states) and solves this problem instead. The solution of the new problem is the optimal value function of the original optimisation problem which is then used to obtain the associated control signals. For further details on HJB the reader is refereed to [91].

Although DP approach finds the global optimal solution of an OCP by providing a sufficient and necessary condition of optimality, yet it suffers from curse of dimensionality which reduces its performance and limits its applications [101]. In other words, because of the fact that the required computing effort to solve an OCP using DP grows exponentially with the number of the system states, DP is often used to solve minor problems or for offline calculation of the benchmark optimal solutions [98].

2.6.2 Indirect method

Indirect method employs Pontryagin's minimum principle (PMP) to convert an OCP to a boundary value ordinary differential problem, and provides the necessary conditions of optimality. PMP claims that if there exists an optimal solution to an OCP, it must also minimise a *Hamiltonian* function $\mathcal{H}$. For example, the Hamiltonian function of the OCP (2.5) ignoring the inequality constraints $h(x, u; t) \leq 0$ is formulated as follows [91]:

$$\mathcal{H}(x_k, u_k, \lambda_{k+1}; k) := L(x_k, u_k; k) + \lambda_{k+1}^T f(x_k, u_k; k). \tag{2.16}$$

where $k \in [0, N_p - 1]$, and λ is the vector of Lagrange multiplier whose elements are the costates of the system and indicate the marginal relaxation in the constraints.

PMP introduces necessary conditions for the control sequence $u^*(.)$ to be optimal which are based on Euler-Lagrange equation. These necessary conditions for an OCP are presented as follows [91]:

$$\mathcal{H}(x_k^*, u_k^*, \lambda_{k+1}^*; k) \leq \mathcal{H}(x_k^*, u_k, \lambda_{k+1}^*; k) \tag{2.17a}$$

$$x_{k+1}^* = \frac{\partial \mathcal{H}}{\partial \lambda_{k+1}}(x_k^*, u_k^*, \lambda_{k+1}^*; k) \tag{2.17b}$$

$$\lambda_k^* = \frac{\partial \mathcal{H}}{\partial x}(x_k^*, u_k^*, \lambda_{k+1}^*; k) \tag{2.17c}$$

$$\lambda_N^* = \frac{\partial \mathcal{M}}{\partial x_N}(x_N^*; N) \tag{2.17d}$$

$$x_k(0) = x_0 \tag{2.17e}$$

$$\forall u(k) \in \mathcal{U}, k \in [0, N-1]. \tag{2.17f}$$

where (2.17d) and (2.17e) are the boundary conditions. The necessary condition in (2.17a) indicates that an optimal control $u^*(.)$ must minimise the Hamiltonian. Also, to minimise the Hamiltonian the following necessary condition must be satisfied:

$$\frac{\partial \mathcal{H}}{\partial u}(x_k^*, u_k^*, \lambda_{k+1}^*; k) = 0. \tag{2.18}$$

Equation (2.18) is a first order necessary condition for optimality and its solution $u^*(k) = \pi(x^*, \lambda_{k+1}^*; k)$ (where π is a control policy function) is a local optimum in terms of x^* and λ^* for the original OCP. Substituting this solution to the rest of conditions results in a boundary value problem which can then be solved using a gradient-based or shooting approach. It is worth noting that, the calculated optimal control input $u^*(.)$ is guaranteed to be a local optimum solution of PMP and hence the OCP if the following second order sufficient condition is satisfied:

$$\frac{\partial^2 \mathcal{H}}{\partial u^2}(x_k^*, u_k^*, \lambda_{k+1}^*; k) > 0. \tag{2.19}$$

In contrast to DP method, indirect method is less computationally demanding [102]. PMP converts an OCP into boundary value problems with ordinary differential equations (ODEs) whereas DP involves solving PDEs which are more computationally expensive. Moreover, the computational burden of solving OCPs using DP increases exponentially with respect to the dimension of states while it is linear for PMP. Although PMP is in general a faster technique than DP, it comes at a cost of not guaranteeing the global optimality of the solution. In other words, the solution calculated by the PMP method is not necessarily a global optimum solution and can be a local one [102].

2.6.3 Direct methods

Solving large-scale OCPs with several equality and inequality constraints based on the necessary and sufficient conditions of optimality of DP or PMP is not always practical. DP suffers from curse of dimensionality when the problem involves many states and PMP does not guarantee the optimality of the solution and has difficulties for solving singular control problems [5]. These limitations of the previous approaches promoted the direct methods to become a more popular approach for solving OCPs.

The direct methods are a class of techniques for solving OCPs which employ the discretise then optimise (D-O) strategy. In contrast to the indirect method, this approach benefits from a systematic handling of the inequality constraints which makes it applicable to a wide range of problems [103].

Direct methods are divided into two groups based on their structure: sequential and simultaneous strategies. In the sequential strategy, there is an inner solver that deals with integration of differential algebraic equations (DAEs) which define the constraints, and an optimiser that use the results of the integration to modify the discretised control signals. Thus the integration and optimisation are performed sequentially and one after the other. Therefore, in this strategy, the only degree of freedom for the NPL solver is the control signals [103] which are discretised as piecewise polynomial, and hence the optimisation is carried out with respect to the polynomial coefficients [104].

A summary of numerical methods for solving OCPs is provided in Fig 2.10.

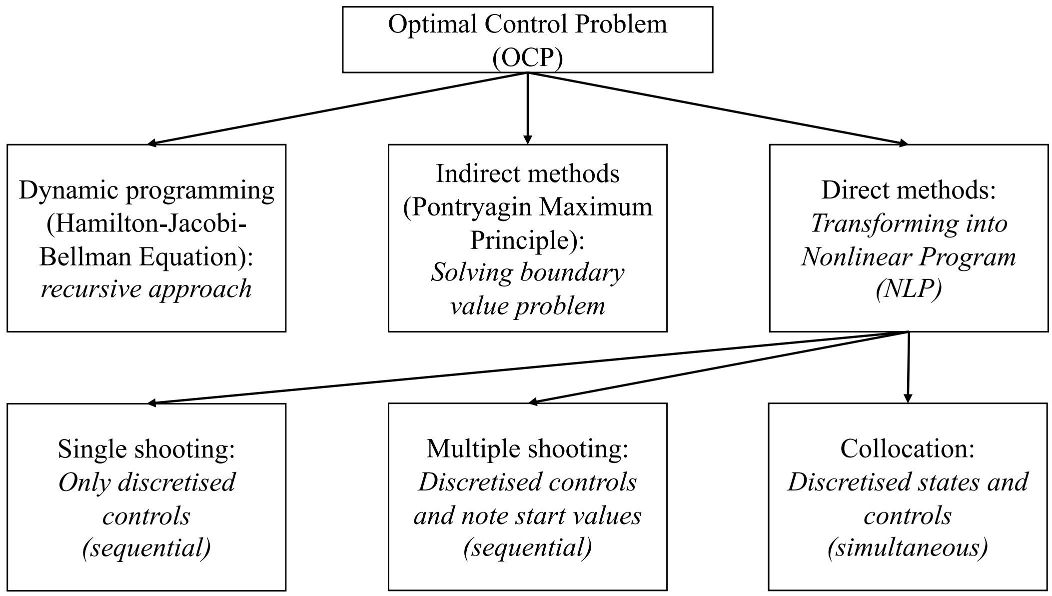


Figure 2.10: An overview of numerical methods for solving optimal control problems.

Direct sequential methods

Fig. 2.11 shows an outline of a sequential strategy called direct single shooting. From Fig. 2.11, it can be seen that the DAE constraints are realised by an inner level DAE solver which integrates the differential equations with respect to the control inputs. The DAE solver, solves an initial value problem and outputs state trajectories. The resulting state trajectories are employed to obtain the gradient of the constraints $\nabla_u h(0)$ and cost function $\nabla_u J(0)$, both with respect to the control inputs. Using these information, the NLP solver updates the control signals and applies them to the DAE solver to iteratively converge to optimal solution. The implementation of direct single shooting method is relatively simple due to the availability of effective NLP and DAE solvers. However, this method is not advised to be used for unstable systems [5, 103]. Moreover, sequential methods require repeated integration of DAEs which can be time consuming for large-scale problems.

Multiple shooting method is developed to improve the performance of single shooting in terms of applicability to unstable systems [105, 106]. The multiple shooting method, as compared to the single shooting, divides the time domain of the system into smaller segments and in each segment the DAE model of the system is integrated separately. To ensure that the solution function is a single continuous function, equality constraints in the form of initial values of states for each time point are added to the NLP [107]. Such equality constraints improve convergence to an optimal solution for unstable systems.

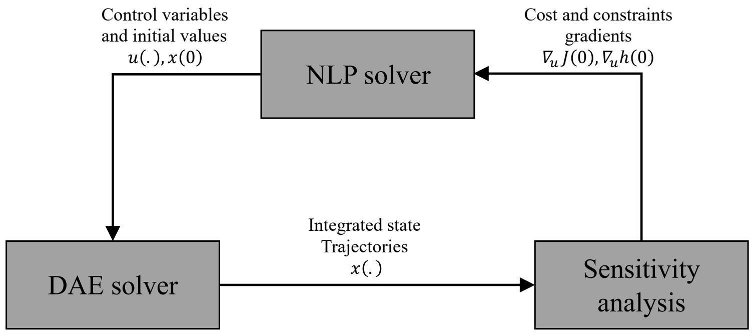


Figure 2.11: Architecture of sequential optimisation method [Courtesy of [5]].

Direct simultaneous (collocation) method

The direct simultaneous optimisation strategy discretises both the control inputs and the states of the system by collocation on finite elements. The discretisation removes the differential equations of the OCP and converts the problem into a large-scale nonlinear programming problem with a set of algebraic equations which can then be solved using an NLP solver. This direct collocation strategy is fully simultaneous, it means that there is no sensitivity calculator or inner DAE solver as opposed to the shooting strategies [108], and in fact the DAE constraints are solved at the optimal solution.

The major aspect of direct simultaneous method is that any NLP solver employed, can benefit from the sparsity structure of the resulting Karush-Kuhn-Tucker (KKT) conditions. This is also the major reason that direct collocation method is employed in this thesis to solve the OCP of CAVs crossing intersection. This technique fully discretises the OCP, leading to a spares NLP which can be exploited by an interior-point method as the NLP solver [109]. Moreover, as compared to sequential methods direct simultaneous method does not require repeated numerical integration of the DAE model which is time consuming particularly for large-scale problems [104, 110]. Moreover, the speed of calculating the gradients and Hessians plays a crucial role in the performance of simultaneous method which are effectively improved using advanced techniques like Automatic Differentiation (AD). Further details on NLP and AD can be found in respectively, Sections 2.6.4 and 2.6.6.

Consider the following ordinary differential equation as the plant dynamics:

$$\dot{x}(t) = f(x, u; t). \tag{2.20}$$

Using the collocation method, the ODE presented in (2.20) is solved at selected points in time. The state variable $x(t)$ is approximated by Lagrange interpolation polynomial of degree K. The Lagrange interpolation polynomials is the preferred representation among the several others because the polynomial coefficients have the same variable bounds as the profiles themselves [5]. A Lagrange polynomial of degree K is generated by $K+1$ interpolation points within each time step i, that construct the following time grid:

$$t = t_{i-1} + h_i\tau, \qquad t \in [t_{i-1}, t_i],\ \tau \in [0, 1]. \tag{2.21}$$

where h_i is the sampling time at i and τ represents the interpolation points.

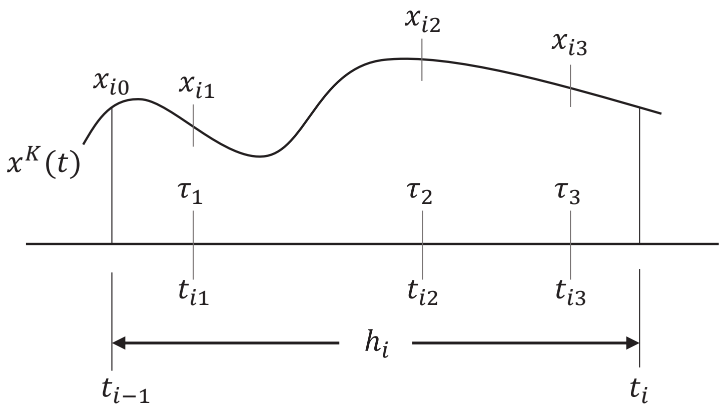


Figure 2.12: Polynomial approximation of the states over a finite element (courtesy of [4]).

Fig. 2.12 shows the Lagrange interpolation with collocation point degree $K = 3$ for the state trajectory of the ODE given by (2.20). The state trajectory in Figure 2.12 at time step i can be approximated with the following Lagrange polynomial $x^K(\tau)$ of degree K [5]:

$$x^K(\tau) = \sum_{j=0}^{K} \mathcal{L}_j(\tau)x_{ij}. \tag{2.22}$$

$$\mathcal{L}_j(\tau) = \prod_{k=0,\neq j}^{K} \frac{\tau - \tau_k}{\tau_j - \tau_k}. \tag{2.23}$$

where τ_j, $j = 0..K-1$ monotonically ascends with time, and $\tau_0 = 0$.

The approximated trajectory (2.22) is substituted into the original ODE (2.20) which after discretisation results in the following collocation equation for each time element i [5]:

$$\begin{aligned}
\dot{x}^K(\tau_k) &= \frac{d}{dt}\sum_{j=0}^{K}\mathcal{L}_j(\tau_k)x_{ij}, \\
&= \frac{1}{h_i}\sum_{j=0}^{K}\frac{d\mathcal{L}_j}{d\tau}(\tau_k)x_{ij}, \\
&\approx f(x_{ik}, u_{ik}), \\
\therefore \qquad \sum_{j=0}^{K}\dot{\mathcal{L}}_j(\tau_k)x_{ij} &\approx h_i f(x_{ik}, u_{ik}), \qquad k = 0, .., K-1.
\end{aligned} \tag{2.24}$$

where h_i is the sampling time at i, and u_{ik} is the discretised control signal using the same concept:

$$u(\tau) = \sum_{j=1}^{K}\tilde{\mathcal{L}}_j(\tau)u_{ij}.$$

where

$$\tilde{\mathcal{L}}_j(\tau) = \prod_{k=1,\neq j}^{K}\frac{\tau-\tau_k}{\tau_j-\tau_j}. \tag{2.25}$$

It is worth nothing that in order to transform an OCP into an NLP, the interpolation points τ_k must be carefully chosen to obtain the best approximation of state variables x. In this regard, it is shown that the best interpolation points τ_k are the roots of a system of orthogonal polynomials $P_K(\tau)$ (K is the maximum degree of the system) [5], that means:

$$\int_0^1 P_k(\tau)P_{k\prime}(\tau)\, d\tau = 0, \qquad \forall k \neq k\prime, \qquad k = 0, .., K-1, \qquad k\prime = 1, ..K. \tag{2.26}$$

The Gauss–Legendre and Radau polynomials are amongst the candidate that have orthogonality property and can be used for finding the collocation points τ_k. Table 2.3 shows the roots of Gauss–Legendre and Radau system of polynomials as collocation points for different values of K.

Table 2.3: Shifted Gauss–Legendre and Radau roots as collocation points.

Degree K	**Legendre Roots**	**Radau Roots**
1	0.500000	1.000000
2	0.211325	0.333333
	0.788675	1.000000
3	0.112702	0.155051
	0.500000	0.644949
	0.887298	1.000000
4	0.069432	0.088588
	0.330009	0.409467
	0.669991	0.787659
	0.930568	1.000000
5	0.046910	0.057104
	0.230765	0.276843
	0.500000	0.583590
	0.769235	0.860240
	0.953090	1.000000

Now that both the states and control signals are fully discretised, it is important to make sure the states are continuous across time elements and collocation points. The following equations enforce continuity of states for N_p time elements:

$$x_{i+1,0} = \sum_{j=0}^{K} \mathcal{L}_j(1) x_{ij}, \qquad i = 1, .., N_p - 1 \tag{2.27a}$$

$$x(t_f) = \sum_{j=0}^{K} \mathcal{L}_j(1) x_{Nj}, \qquad x_{1,0} = x(t_0). \tag{2.27b}$$

where $x(t_0)$ and $x(t_f)$ are given initial and final boundaries.

Using the collocation and continuity equations, the original OCP_{N_p} (2.5) is discretised and transformed into an NLP which can then be solved using NLP solvers. The resulting NLP formulation is given as follows [4]:

$$u^* = \underset{u \in \mathcal{U}^{N_p+K}}{\text{arg minimise}} \quad J_{N_p}(x, u; n) := \sum_{i=n}^{n+N_p-1} \sum_{j=0}^{K} \mathcal{L}(x_{ij}, u_{ij}) \tag{2.28a}$$

subject to:

$$\sum_{j=0}^{K} \dot{\mathcal{L}}_j(\tau_k) x_{ij} - h_i f(x_{ik}, u_{ik}) = 0, \tag{2.28b}$$

$$h(x_{ij}, u_{ij}) \leq 0, \tag{2.28c}$$

$$x_{i+1,0} = \sum_{j=0}^{K} \mathcal{L}_j(1) x_{ij}, \tag{2.28d}$$

$$x(t_f) = \sum_{j=0}^{K} \mathcal{L}_j(1) x_{(n+N_p)_j}, \quad x_{n,0} = x(t_0), \tag{2.28e}$$

$$g(x(n + N_p)) = 0, \tag{2.28f}$$

$$i \in \{n + 1, .., n + N_p - 1\}, \quad k \in \{1, .., N_p \times K\}, \tag{2.28g}$$

$$x_{ij} \in X, u_{ij} \in U. \tag{2.28h}$$

2.6.4 Non-linear programming (NLP)

OCPs are converted into NLPs by employing a direct method such as the simultaneous collocation strategy. The resulting large-scale NLP with both the equality and inequality constraints is as follows [111]:

$$\underset{x \in \mathbb{R}^n}{\text{minimise}} \quad f(x) \tag{2.29a}$$

subject to:

$$h(x) \leq 0, \tag{2.29b}$$

$$g(x) = 0. \tag{2.29c}$$

where (2.29b) and (2.29c) represent, respectively, the vector of inequality and equality constraints of the NLP formulation.

In order to solve the NLP proposed in (2.29) there exists at least three methods that can handle both the equality and inequality constraints [5]: interior-point methods (IPMs), sequential quadratic programming (SQP) methods and nested projection methods. There is an extensive comparison between the NLP solvers and the reader is referred to [112]. The interior-point method is selected as the NLP solver in this study and an overview of this technique is presented in the next section.

2.6.5 Interior-point method (IPM)

The interior-point method solves NLPs, given as in (2.29), by initially relaxing the problem then employing fast iterative solvers to solve the relaxed version. IPOPT as an open-source implementation of IPM solves large-scale NLPs efficiently. This study employs IPOPT as the NLP solver, majorly because the sparse NLPs generated from fully discretised OCPs, when solved by IPOPT lead to a sparse banded KKT matrix. The KKT matrix is a large system of multi-variable algebraic equations which can be solved fast using lower-upper factorisation based linear solvers such as MA57 [113] or MUMPS [114]. A detailed explanation of structure exploitation in an interior-method for fully discretised OCPs can be found in [109].

The main strategy of IPM is to convert the inequality constraints of the original problem into relaxed equality constraints using either of the *barrier* or *homotopy* approaches.

The homotopy approach converts the inequality constraints of the NLP (2.29) into equality constraints using a vector of slack variables s as follows:

$$\underset{x \in \mathbb{R}^n, s \in \mathbb{R}^{n_s}}{\text{minimise}} \quad f(x) \tag{2.30a}$$

subject to:

$$h(x) - S = 0, \tag{2.30b}$$

$$g(x) = 0. \tag{2.30c}$$

where the number of equality constraints is denoted as n_s and $S = diag(s)$.

The corresponding Lagrangian function is defined as:

$$\mathscr{L}(x, \lambda, \mu) = f(x) - g(x)^T \lambda - (h(x) - S)^T \mu \tag{2.31}$$

Base on the Lagrangian function, the KKT theorem is employed to construct necessary conditions of optimality as the following set of algebraic equations:

$$\nabla_x \mathscr{L}(x^*, \lambda^*, \mu^*) = 0, \tag{2.32a}$$

$$S^* \mu^* - \upsilon e = 0, \tag{2.32b}$$

$$g(x^*) = 0, \tag{2.32c}$$

$$h(x^*) - S^* = 0, \tag{2.32d}$$

$$S^* \succeq 0. \tag{2.32e}$$

where $e = [1, 1, .., 1]^T$ and υ is a perturbation parameter that vanishes iteratively and enforces the solution to keep distance from boundaries, i.e., $\lim_{i\to\infty} \upsilon_i = 0$.

Fast iterative approaches such as trust-region [115] or linesearch [40] methods can be employed to efficiently solve problem (2.32) to obtain the solution $(x^*, \lambda^*, \mu^*, S^*)$ iteratively for a vanishing value of υ.

The other interior-point based approach is called *barrier* method that adds a log-barrier term to the cost function to construct a relaxed version of the original NLP (2.30) as follows:

$$\underset{x\in\mathbb{R}^n, \upsilon_l\in\mathbb{R}^n}{\text{minimise}} \quad f(x) - \upsilon_l \sum_{i=1}^{n_h} ln(-s_i) \tag{2.33a}$$

subject to:

$$h_i(x) - s_i = 0, \forall i = 1..n_h \tag{2.33b}$$

$$g(x) = 0. \tag{2.33c}$$

where n_h represents the number of inequality constraints, υ_l denotes the barrier parameter and l is an integer presenting the sequence counter such that $\lim_{l\to\infty} \upsilon_l = 0$. It should be noted that the log-barrier function is unbounded at $h(x) = 0$ and therefore, the inequality constraint $h(x)$ is strictly positive. A local solution of the original NLP is obtained by iteratively solving the relaxed problem (2.33) so that the barrier parameter $\{\upsilon_l\}$ converges zero. Hence, the barrier method solves NLPs by constructing a barrier problem such as the one in (2.33) and as the value of the barrier parameter υ_l decreases the solution of (2.33) converges to a solution of the original NLP. To solve the barrier problem, a similar iterative solver as the *homotopy approach*, such as a Newton based method is employed to iteratively calculate the formulated KKT conditions which leads to the following sparse linear problem at each iteration i that finds the system variables ($x_{i+1} = x_i + \triangle x, \lambda_{i+1} = \lambda_i + \triangle\lambda, \upsilon_{i+1} = \upsilon_i + \triangle\upsilon$):

$$\begin{bmatrix} Q_i & A_i & -I \\ A_i^T & 0 & 0 \\ X_i & 0 & X_i \end{bmatrix} \begin{bmatrix} \triangle x \\ \triangle\lambda \\ \triangle\upsilon \end{bmatrix} = - \begin{bmatrix} \triangledown f(x_i) + A_i\lambda_i - \upsilon_i \\ g(x_i) \\ X_i D_i e - \mu_l e \end{bmatrix} \tag{2.34}$$

where $e = [1, 1, .., 1]^T$, the diagonal elements are as $X_i = diag(x_i)$ and $D_i = diag(\upsilon_i)$, the Hessian is denoted by $Q_i = \triangledown_{xx}\mathscr{L}(x_i, \lambda_i, \upsilon_i)$ and $A_i = \triangledown g(x_i)$.

2.6.6 Automatic differentiation (AD)

The performance of algorithms for solving NLPs, and hence OCPs significantly depends on the precision and speed of the method employed to calculate Jacobian and Hessian matrices. Therefore, differentiation is a fundamental property of NLP solvers and is one of the major barriers to the accuracy and performance of NLPs.

AD is a simple but extremely useful technique for fast and accurate calculation of derivative of any degrees. AD as compared to symbolic differentiation does not lead to inadequate codes for complex equations where high degree of derivative degree is required and as compared to numerical differentiation, it does not arise rounding errors. Lastly, AD resolves the issue of both of these classical approaches in terms of speed of differentiating partial derivatives of a function with respect to many inputs, as it is required for the gradient-based optimisation algorithms.

The fundamental idea of AD is that every computer program (function), no matter how complicated, performs a set of elementary arithmetic operations. AD evaluates the derivative of such a function defined by a computer program, by decomposing the elementary operations and developing the derivative of each operation. The overall derivative of the function is then calculated by another computer program that uses the chain rule and intermediate variables to each elementary operation. There exist two distinct modes of AD namely, *forward* mode and *reverse* mode. In the forward mode AD, decomposition is performed from the inner operations to the outer ones whereas this is the other way around for the reverse mode AD.

Table 2.4 shows the steps of forward mode AD for a sample function $y = (x + x^3)^3$. The numerical values of the derivatives and values of intermediate variables for the sample function are determined by seeding the variable x as 4 and its derivative as 2.

On the other hand, Table 2.5 shows the operation of AD in the reverse mode for the same sample function. In the reverse mode, $\underline{\omega}$ denotes the derivative of output y with respect to ω hence $\underline{y} = 2$. Similar to the previous mode, the numerical values of the derivatives and intermediate values are determined by seeding the variable x as 4.

It is shown that AD saves considerable amount of computational time for calculating Hessian and Jacobians as compared to numerical and symbolic methods [116]. The reader is referred to [117, 118] for more information about AD.

Table 2.4: Automatic differentiation in the forward mode for a sample of function $y = f(x) = (x + x^3)^3$.

Elementary operations	Forward derivatives	Intermediate values	Derivatives values
$\omega_0 = x = 4$	$\dot{\omega_0} = \dot{x} = 2$	4	2
$\omega_1 = x^3 = \omega_0^3$	$\dot{\omega_1} = 3\omega_0\dot{\omega_0}$	64	24
$\omega_2 = x + x^3 = \omega_0 + \omega_1$	$\dot{\omega_2} = \dot{\omega_0} + \dot{\omega_1}$	68	26
$\omega_3 = (x + x^3)^3 = \omega_2^3$	$\dot{\omega_3} = 3\omega_2\dot{\omega_2}$	314432	5304

Table 2.5: Automatic differentiation in the reverse mode for a sample of function $y = f(x) = (x + x^3)^3$.

Elementary operations	reverse derivatives	Intermediate values	Derivatives values
$\omega_3 = (x + x^3)^3 = \omega_2^3$	$\underline{\omega_3}$	314432	2
$\omega_2 = x + x^3 = \omega_0 + \omega_1$	$\underline{\omega_2} = 3\omega_2\underline{\omega_3}$	68	408
$\omega_1 = x^3 = \omega_0^3$	$\underline{\omega_1} = \underline{\omega_2}$	64	408
$\omega_0 = x = 4$	$\underline{\omega}_0 = (1 + 3\omega_0)\underline{\omega_2}$	4	5304

2.6.7 Toolkits

To effortlessly formulate and solve OCPs, there exists a number of toolkits that are implemented in MATLAB, C++/C and FORTRAN. This study employs computer algebra system with automatic differentiation implementation (CasADi). CasADi is an open-source framework that allows to flexibly formulate and solve OCPs and MPCs. This toolkit includes interfaces for solving NLPs and DAE as well as the implementation of forward and inverse AD [119]. In addition, CasADi is a computer algebra system (CAS) and allows users to choose different direct methods such as multiple and single shooting methods or collocation method to solve the OCPs. The implementation of each one of these direct methods can be designed by the user to be the most suitable for the problem.

As compared to other toolkits such as ACADO [120], FORCESPRO [121] and MUSCOD-II [106] that only allow the implementation of single and multiple shooting methods for discretisation of differential equations, CasADi allows the implementation of the collocation method. The single and multiple shooting methods are based the Euler method while the collocation method is based on Lagrange polynomial approximation. The key difference between the two discretisation methods is that collocation method uses a polynomial of higher degree to approximate the solution, while the Euler method uses a linear approximation. As a result, the collocation method is generally more accurate than the Euler method for the same number of function evaluations, but it is also more computationally expensive. Moreover, unlike the mentioned toolkits that only provide the user with a black box OCP solver, CasADi is a flexible tool that supports low-level implementation which is useful for formulating any desired OCP. For additional information on CasADi and its applications, the reader is referred to [117–119].

Chapter 3

Optimal Lane-Free Crossing of CAVs through Intersections

This chapter shows the formulation and solution of lane-free crossing of CAVs through intersections as a minimum-time optimal control problem that minimises the crossing time as well as the energy consumption due to acceleration of all CAVs. Initially, the shape of vehicles and road boundaries are modelled in a way that lane-free movement is be enabled. Thereafter, a smoothing technique based on dual problem theory of convex optimisation is employed to smoothen the constraints of CAVs avoiding collision with each other and with road boundaries. A smooth minimum-time OCP is then formulated to achieve the shortest crossing time while minimising the energy consumption of all vehicles passing through an intersection in a lane-free order. Finally, the performance of the proposed lane-free algorithm in terms of crossing time, energy consumption and passenger comfort is compared against two state-of-the-art algorithms where one is based on a similar concept of lane-free crossing and the other is a reservation-based method.

3.1 Modelling the Shape of Vehicles and Road Boundaries

This study represents each CAV$_i$, when $i \in \{1..N_v\}$ and N_v is the total number of CAVs, as a rectangular polytope $\tilde{\mathcal{P}}_i$ (i.e., a convex set) that is the intersection area of half-space linear inequalities $\tilde{\mathbf{A}}_i\mathbf{x} \leq \tilde{\mathbf{b}}_i$ at the origin, where $\mathbf{x} \in \mathbb{R}^2$ is a Cartesian point. In this paper, all CAVs have the same size which are defined with:

$$\tilde{\mathbf{A}}_i = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}^\top, \quad \tilde{\mathbf{b}} = [l/2, l/2, d/2, d/2]^\top. \tag{3.1}$$

where l and d denote, respectively, the wheelbase and track of CAVs.

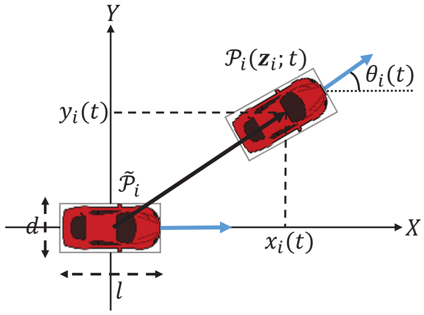


Figure 3.1: Transformation of each CAV$_i$ from $\tilde{\mathcal{P}}_i$ to $\mathcal{P}_i(\mathbf{z}_i;t)$ where $\mathbf{z}_i(t) = [x_i(t), y_i(t), \theta_i(t)]^T$.

As CAV$_i$ moves to a new pose $\mathbf{z}_i(t) = [x_i(t), y_i(t), \theta_i(t)]^T$, the original polytope $\tilde{\mathcal{P}}_i$ of the CAV is transformed to $\mathcal{P}_i$ as follows:

$$\tilde{\mathcal{P}}_i \mapsto \mathcal{P}_i(\mathbf{z}_i;t) : \mathbf{A}_i(\mathbf{z}_i;t)\mathbf{x}(t) \leq \mathbf{b}_i(\mathbf{z}_i;t). \tag{3.2}$$

where:

$$\mathbf{A}_i(\mathbf{z}_i;t) = \tilde{\mathbf{A}}_i \begin{bmatrix} cos\theta_i(t) & sin\theta_i(t) \\ -sin\theta_i(t) & cos\theta_i(t) \end{bmatrix}, \tag{3.3a}$$

$$\mathbf{b}_i(\mathbf{z}_i;t) = \tilde{\mathbf{b}} + \tilde{\mathbf{A}}_i \begin{bmatrix} cos\theta_i(t) & sin\theta_i(t) \\ -sin\theta_i(t) & cos\theta_i(t) \end{bmatrix} [x_i(t), y_i(t)]^\top. \tag{3.3b}$$

Fig. 3.1 provides a graphical representation of (3.2). It is worth noting that the robot pose in (3.3b) (e.g., $x_i(t), y_i(t)$ and $\theta_i(t)$) do not cause non-convexity as the solver treats them as variables and substitutes values there.

Road boundaries are also modelled as convex polytopic sets $\mathcal{O}_r$, when $r \in \{1..N_r\}$ and N_r is the total number of road boundaries which is 4 for four-legged intersections.

Based on these representations, there is no collision between CAV$_i$ and CAV$_j$ if and only if $\mathcal{P}_i(\mathbf{z}_i;t) \cap \mathcal{P}_j(\mathbf{z}_j;t) = \emptyset, \forall t \in [t_0, t_f]$. Similarly, CAVs do not collide with road boundaries when the intersection of their sets is always empty, i.e. $\mathcal{P}_i(\mathbf{z}_i;t) \cap \mathcal{O}_r = \emptyset, \forall t \in [t_0, t_f]$.

3.2 Smoothing of Constraints to Avoid Collisions Between CAVs

To avoid collisions between any CAV$_i$ and CAV$_j$ $\forall i \neq j \in \{1..N_v\}$, their polytopic sets should not intersect, i.e. $\mathcal{P}_i \cap \mathcal{P}_j = \emptyset$ where $\mathcal{P}_i = \{\mathbf{x} \in \mathbb{R}^2 | \mathbf{A}_i\mathbf{x} \leq \mathbf{b}_i\}$ and $\mathcal{P}_j = \{\mathbf{y} \in$

$\mathbb{R}^2|\mathbf{A}_j\mathbf{y} \leq \mathbf{b}_j\}$. However, these are non-differentiable conditions and enforcing them as constraints in an OCP will make the problem difficult to be solved by the state-of-the-art gradient-based algorithms. To preserve differentiability and continuity, $\mathcal{P}_i \cap \mathcal{P}_j = \emptyset$ is replaced by the following sufficient condition which has negligible effect on the optimality of the solution for small values of d_{min} [122]:

$$dist(\mathcal{P}_i, \mathcal{P}_j) = \min_{\mathbf{x},\mathbf{y}}\{\|\mathbf{x} - \mathbf{y}\|_2 \mid \mathbf{A}_i\mathbf{x} \leq \mathbf{b}_i,\ \mathbf{A}_j\mathbf{y} \leq \mathbf{b}_j\} \geq d_{min};$$
$$\forall i \neq j \in \{1..N\}. \tag{3.4}$$

where d_{min} is the minimum safe distance between CAVs.

Problem (3.4) is still non-convex and non-differentiable [122] and the remaining of this subsection is dedicated to reformulate (3.4) with a smooth sufficient condition.

It is known that the problem of finding the minimum distance between two polytopes $\mathcal{P}_i$ and $\mathcal{P}_j$ (the left hand-side of (3.4)) is convex [95]. Also, since $\mathcal{P}_j$ is not an empty set, the strong duality holds [122]. This means that the solution of the primal problem of finding $dist(\mathcal{P}_i, \mathcal{P}_j)$ is the same as the one of its dual problem which is as follows:

$$\begin{aligned} \mathrm{dist}(\mathcal{P}_i, \mathcal{P}_j) := \max_{\boldsymbol{\lambda}_{ij},\boldsymbol{\lambda}_{ji},\mathbf{s}_{ij}} \quad & -\mathbf{b}_i^\top\boldsymbol{\lambda}_{ij} - \mathbf{b}_j^\top\boldsymbol{\lambda}_{ji} \\ \text{s.t.} \quad & \mathbf{A}_i^\top\boldsymbol{\lambda}_{ij} + \mathbf{s}_{ij} = 0, \mathbf{A}_j^\top\boldsymbol{\lambda}_{ji} - \mathbf{s}_{ij} = 0, \\ & \|\mathbf{s}_{ij}\|_2 \leq 1, -\boldsymbol{\lambda}_{ij} \leq 0, -\boldsymbol{\lambda}_{ji} \leq 0; \\ & \forall i \neq j \in \{1..N\}. \end{aligned} \tag{3.5}$$

where $\boldsymbol{\lambda}_{ij}$, $\boldsymbol{\lambda}_{ji} \in \mathbb{R}^4$, and $\mathbf{s}_{ij} \in \mathbb{R}^2$ are the dual variables and $\mathbf{A}_i$ and $\mathbf{b}_i$ are as in (3.3) (the deviation of dual problem (3.5) from primal problem (3.4) is shown in [123]).

Combining (3.5) with (3.4), the objective function of (3.5) subject to its constraints must be greater than or equal to d_{min} in order to avoid collisions. However, (3.5) can be substituted by $\{\exists\boldsymbol{\lambda}_{ij} \geq 0, \boldsymbol{\lambda}_{ji} \geq 0, \mathbf{s}_{ij} : -\mathbf{b}_i^\top\boldsymbol{\lambda}_{ij} - \mathbf{b}_j^\top\boldsymbol{\lambda}_{ji} \geq d_{cmin}, \mathbf{A}_i^\top\boldsymbol{\lambda}_{ij} + \mathbf{s}_{ij} = 0, \mathbf{A}_j^\top\boldsymbol{\lambda}_{ji} - \mathbf{s}_{ij} = 0, \|\mathbf{s}_{ij}\|_2 \leq 1\}$ because the existence of a feasible solution $\boldsymbol{\lambda}_{ij,feas}$, $\boldsymbol{\lambda}_{ji,feas}$, and $\mathbf{s}_{ij,feas}$ where $-\mathbf{b}_i^\top\boldsymbol{\lambda}_{ij,feas} - \mathbf{b}_j^\top\boldsymbol{\lambda}_{ji,feas} \geq d_{min}$ is a sufficient condition to ensure $dist(\mathcal{P}_i, \mathcal{P}_j) \geq d_{min}$, i.e. to avoid collisions [123]. Also, It is shown in [122] that these sufficient conditions are smooth since the norm operator in $\|\mathbf{s}_{ij}\|_2 \leq 1$ is an Euclidean distance as $\mathbf{s}_{ji} \in \mathbb{R}^2$ and the resulting feasible values of $\mathbf{s}_{ij}$ make a quadratic cone. Moreover, the proposed sufficient conditions replace the nested optimisation problem with feasibility inequality constraints.

3.3 Smoothing of Constraints to Avoid Collisions with Road Boundaries

Each CAV$_i$ must also avoid all the road boundaries, i.e. $\mathcal{P}_i \cap \mathcal{O}_r = \emptyset$ where $\mathcal{P}_i = \{\mathbf{x} \in \mathbb{R}^2 | \mathbf{A}_i \mathbf{x} \leq \mathbf{b}_i\}$ and $\mathcal{O}_r = \{\mathbf{y} \in \mathbb{R}^2 | \mathbf{A}_r \mathbf{y} \leq \mathbf{b}_r\}$. Similar to section 3.2 collision avoidance between CAVs, $\mathcal{P}_i \cap \mathcal{O}_r = \emptyset$ is replaced by the following sufficient condition:

$$\begin{aligned} \text{dist}(\mathcal{P}_i, \mathcal{O}_r) = \min_{\mathbf{x},\mathbf{y}} \{\|\mathbf{x} - \mathbf{y}\|_2 \,|\mathbf{A}_i \mathbf{x} \leq \mathbf{b}_i, \mathbf{A}_r \mathbf{y} \leq \mathbf{b}_r\} \geq d_{rmin}; \\ \forall r \in \{1..N_r\}. \end{aligned} \tag{3.6}$$

where d_{rmin} is the minimum safety distance between CAVs and road boundaries.

The dual problem of (3.6) is then substituted with the sufficient condition $\{\exists \boldsymbol{\lambda}_{ir} \geq 0, \boldsymbol{\lambda}_{ri} \geq 0, \mathbf{s}_{ir} : -\mathbf{b}_i^\top \boldsymbol{\lambda}_{ir} - \mathbf{b}_r^\top \boldsymbol{\lambda}_{ri} \geq d_{rmin}, \mathbf{A}_i^\top \boldsymbol{\lambda}_{ir} + \mathbf{s}_{ir} = 0, \mathbf{A}_r^\top \boldsymbol{\lambda}_{ri} - \mathbf{s}_{ir} = 0, \|\mathbf{s}_{ir}\|_2 \leq 1\}$ where $\boldsymbol{\lambda}_{ir}, \boldsymbol{\lambda}_{ri}$, and $\mathbf{s}_{ir}$ are the dual variables. $\mathbf{s}_{ir}$ is the separating hyperplane between CAVs and road boundaries (see Fig. 2.6).

3.4 Formulation of the Problem as an OCP

Lane-free crossing of multiple CAVs through a signal-free intersection is formulated as the following optimal control problem:

$$\{a_i(.), \delta_i(.)\}^* = \tag{3.7a}$$

$$\arg \min_{t_f, a_i(.), \delta_i(.)} J(\mathbf{z}_1(.), .., \mathbf{z}_{N_p}(.)) \tag{3.7b}$$

$$\text{s.t.} \quad (2.1),\ (2.3), \tag{3.7c}$$

$$\mathcal{P}_i(t) \cap \mathcal{P}_j(t) = \emptyset;\ \forall i \neq j \in \{1..N_v\}, \tag{3.7d}$$

$$\begin{aligned} \mathcal{P}_i(t) \cap \mathcal{O}_r(t) = \emptyset;\ \forall i \in \{1..N_v\}, \\ \forall r \in \{1..N_r\}, \end{aligned} \tag{3.7e}$$

$$\mathbf{z}_i(t_0) = \mathbf{z}_{i,0},$$

$$\mathbf{z}_i(t_f) = \mathbf{z}_{i,N_p},$$

$$\forall i \in \{1..N_v\},\ t \in [t_0, t_f].$$

where $J(\mathbf{z}_1(.), .., \mathbf{z}_{N_p}(.))$ is the objective function of the OCP and is defined in the next section 3.5, (3.7c) refers to the vehicle kinematics and (3.7d) and (3.7e) denote, respectively, collision avoidance constraints of each CAV with others and with road boundaries.

As discussed in sections 3.2 and 3.3, the non-differentiable and non-convex collision avoidance constraints (3.7d) and (3.7e) are substituted by the dual problem of their sufficient conditions (3.4) and (3.6), and then (3.7) is reformulated as the following smooth and continuous problem, which is solvable by the state-of-the-art gradient-based algorithms:

$$
\begin{aligned}
&\{a_i(.), \delta_i(.)\}^* = && (3.8a)\\
&\quad \arg \min_{\substack{t_f, a_i(.), \delta_i(.)\\ \boldsymbol{\lambda}_{ij}, \boldsymbol{\lambda}_{ji}, \mathbf{s}_{ij},\\ \boldsymbol{\lambda}_{ri}, \boldsymbol{\lambda}_{ir}, \mathbf{s}_{ir}}} J(\mathbf{z}_1(.), .., \mathbf{z}_{N_p}(.)) &&\\
&\quad \text{s.t.} \quad (2.1), (2.3), && (3.8b)\\
&\qquad -\mathbf{b}_i(\mathbf{z}_i(t))^\top \boldsymbol{\lambda}_{ij}(t) - \mathbf{b}_j(\mathbf{z}_j(t))^\top \boldsymbol{\lambda}_{ji}(t) \geq d_{min}, && (3.8c)\\
&\qquad \mathbf{A}_i(\mathbf{z}_i(t))^\top \boldsymbol{\lambda}_{ij}(t) + \mathbf{s}_{ij}(t) = 0, && (3.8d)\\
&\qquad \mathbf{A}_j(\mathbf{z}_j(t))^\top \boldsymbol{\lambda}_{ji}(t) - \mathbf{s}_{ij}(t) = 0, && (3.8e)\\
&\qquad -\mathbf{b}_i(\mathbf{z}_i(t))^\top \boldsymbol{\lambda}_{ir}(t) - \mathbf{b}_r^\top \boldsymbol{\lambda}_{ri}(t) \geq d_{rmin}, && (3.8f)\\
&\qquad \mathbf{A}_i(\mathbf{z}_i(t))^\top \boldsymbol{\lambda}_{ir}(t) + \mathbf{s}_{ir}(t) = 0, && (3.8g)\\
&\qquad \mathbf{A}_r^\top \boldsymbol{\lambda}_{ri}(t) - \mathbf{s}_{ir}(t) = 0, && (3.8h)\\
&\qquad \boldsymbol{\lambda}_{ij}(t),\ \boldsymbol{\lambda}_{ji}(t),\ \boldsymbol{\lambda}_{ir}(t),\ \boldsymbol{\lambda}_{ri}(t) \geq 0, && (3.8i)\\
&\qquad \|\mathbf{s}_{ij}(t)\|_2 \leq 1,\ \|\mathbf{s}_{ir}(t)\|_2 \leq 1, && (3.8j)\\
&\qquad \mathbf{z}_i(t_0) = \mathbf{z}_{i,0}, && (3.8k)\\
&\qquad \mathbf{z}_i(t_f) = \mathbf{z}_{i,N_p}, && (3.8l)\\
&\qquad \forall i \neq j \in \{1..N_v\}, \forall r \in \{1..N_r\}. &&
\end{aligned}
$$

where $\mathbf{A}_i$ and $\mathbf{b}_i$ are functions of each CAV's pose $\mathbf{z}_i(t)$, and present CAV$_i$ polytope at each time step t. Problem (3.8) is solved at time t_0 for N_p CAVs until the terminal time t_f. The solution to this problem is optimal trajectories of the control signals $a_i(.)^*$ and $\delta_i(.)^*$ of each CAV$_i$ for each $t \in [t_0, t_f]$, as well as a terminal time t_f. CAVs follow their calculated trajectories to arrive their final destinations at the terminal time t_f.

The initial pose $\mathbf{z}_i(t_0)$, i.e. initial position, heading angle and initial speed of all CAV$_i$ $\forall i \in \{1..N_v\}$ within the control zone are known. The remaining of the states and the initial inputs to the CAVs are also assumed as zero. These initial conditions at $t = t_0$

are feasible solutions of the OCP.

3.5 The Objective Function of the OCP

CAVs are expected to reach their terminal pose as fast as possible while consume energy (due to acceleration) as little as possible. Therefore, this study proposes the objective function (3.9) that minimises the overall crossing time of all CAVs and the error between the current and final pose, as well as the energy consumption due to acceleration of each vehicle:

$$J(\mathbf{z}_1(.), .., \mathbf{z}_{N_p}(.)) = \alpha(t_f - t_0)^2 + \mathbf{z}_{i,N_p} \tag{3.9}$$
$$\int_{t_0}^{t_f} \sum_{i=1}^{N_p-1} [(\mathbf{z}_i(t) - \mathbf{z}_i(t_f))^\top \mathbf{Q}(\mathbf{z}_i(t) - \mathbf{z}_i(t_f)) + \gamma a_i(t)^2] \, dt \, .$$

where α, $\mathbf{Q}$ and γ are the gain factors related to the crossing time, CAVs' pose and energy consumption (due to acceleration) respectively. The gains are selected based on trial and error to best normalise the cost function. The expression $(t_f - t_0)^2$ minimises the crossing time of all CAVs. The Lagrange term penalises the error between the current pose $\mathbf{z}_i(t)$ and the final pose $\mathbf{z}_i(t_f)$ as well as the acceleration $a_i(t)^2$ of vehicles. The final pose of CAVs $\mathbf{z}_i(t_f)$ is directly imposed in the objective function and indicates the intended destination of each CAV_i.

3.6 Solving the Proposed Lane-Free Algorithm

To solve the proposed OCP(3.8) ACADO toolkit [120] was initially used with the multiple shooting approach for discretising the dynamics. However, the ACADO toolkit was unable to solve the problem due to potential reasons including non-convexity of the problem and non-linearity of the dynamics, as well as the minimum-time aspect of the OCP. Minimum-time OCP problems struggle with more complexity as the final time of the process is also a decision variable which is multiplied to other variables.

To solve the minimum-time OCP (3.8), this study developed a solver in CasADi [124]. As compared to ACADO which receives the problem formulation and handles the rest, CasADi is a flexible low-level toolkit to develop proprietary solvers for OCPs by providing tool set to discretise system dynamics with collocation method and the automatic differentiation techniques to efficiently calculate Jacobian and Hessian matrices. This flexibility helps to develop proprietary solvers for complex nonlinear problems. Main parts of the developed solver are explained in Appendix A. More details on the CasADi toolkit is provided in Sections 2.6.7 and 2.6.6.

As explained in Section 2.6.5, IPOPT is chosen since it can effectively solve the resulting sparse and banded KKT equations generated from discretising OCPs. To improve the computation time, this study linked IPOPT to Intel® oneAPI Math Kernel Library (oneMKL, https://software.intel.com), which includes high-performance implementation of the *MA27* linear solver. The choice of linear solver is crucial for solving NLPs which are translated to a large linear system of equations using KKT criteria. This study shows that solving the OCP (3.8) with *MUMPS* as a common linear solver, is twice more computationally expensive than MA27. For more explanation on the choice of linear solver MA27 the reader is referred to Section 2.6.5.

All the provided results in this study have been generated by MATLAB R2020 running on a Linux Ubuntu 20.04.0 LTS server with a 3.7 GHz Intel® Core i7 and 32 GB of memory. CasADi was being used through its MATLAB interface.

3.7 Effectiveness of the Proposed Lane-Free Algorithm

In this section, performance of the proposed algorithm is compared against two state-of-the-art benchmarks in terms of crossing time, energy consumption due to acceleration and passenger comfort. For doing this, this study employs the intersection scenario proposed in [1], which is named test scenario one hereafter. The first benchmark is a conflict-point-reservation approach presented in [1], where each CAV calculates its own trajectory by jointly minimising the travelling time and energy consumption (due to acceleration). The calculated reservation times for each conflict point are then shared with other vehicles through a centralised coordinator. Vehicles entering the intersection later read these reserved times and treat them as additional collision avoidance constraints when they plan their own trajectory. The second benchmark is a lane-free method proposed in [2] where CAVs can freely use all the space of the junction, as long as there is no collision. The proposed algorithm in [2] calculates the control inputs for a relatively large given value of crossing time.

Table 3.1: Main parameters of the model

Parameter(s)	Description	Value(s)
m (kg)	mass of each CAV	1204
d_{min} (m)	minimum distance between CAVs	0.1
$d_{rmin}(m)$	minimum distance between CAVs and road boundaries	0.1
d (-)	number of collocation points	5
N_p (-)	number of control intervals	30
$\bar{V}$ (m/s)	upper bound on V_i	25
$\underline{V}$ (m/s)	lower bound on V_i	0
$\bar{\delta}$ = -$\underline{\delta}$ (rad)	bounds on $\lvert\delta_i\rvert$	0.67
$\bar{a} = -\underline{a}$ (m/s^2)	bounds on $\lvert a_i\rvert$	3
$\bar{r}$ = -$\underline{r}$ (rad/s)	bounds on $\lvert r_i\rvert$	0.7
$\bar{\beta}$ = -$\underline{\beta}$ (rad)	bounds on $\lvert\beta_i\rvert$	0.5
$V_i(t_0)$ $\forall i \in \{1..N_v\}$ (m/s)	initial speed of CAVs	10

The algorithms are compared within test scenario one in terms of crossing time, average and standard deviation of speed and energy consumption (due to acceleration) for different number of CAVs between 2 to 12. The initial and terminal pose of CAVs are chosen randomly and there exists at least one CAV performing a left-turn manoeuvre in each test. Table 3.1 summarises values of the parameters used in the proposed and benchmark algorithms. The vehicle parameters and boundaries are typical values for a passenger car. It is assumed that the vehicles only move forward. The number of control intervals N_p and number of collocation points d are tuned to get the best performance with the minimum computational time.

Furthermore, the performance of the proposed algorithm is analysed for two more complex scenarios, which are named test scenario two and test scenario three hereafter. These two scenarios involve up to 21 CAVs, and allow any travelling direction by CAVs (e.g., right, straight and left). The proposed algorithm is also tested for 30 random scenarios for different numbers of CAVs to ensure the statistical significance of the results.

3.7.1 Effectiveness in terms of Crossing Time

This section compares the minimum crossing time of CAVs that can be achieved by the developed and benchmark algorithms. The acceleration gain γ in (3.9) is set to zero to calculate the minimum-time travelling trajectories of CAVs. In other words, the energy consumption due to acceleration is not considered and CAVs only try to reach destinations as fast as possible, which makes the problem single objective.

Table 3.2 compares crossing time of CAVs when they are controlled by the developed and benchmark algorithms during test scenario one. The table also summarises energy consumption due to acceleration, the travelled distance and average and standard deviation of speed of CAVs. It is worth noting that the crossing time is defined as the time required for all the under-control CAVs to cross the intersection and arrive to their destinations. Also, the travelled distance and energy consumption are calculated for all the crossing CAVs.

As seen in Table 3.2, crossing time of CAVs when they are controlled by the proposed algorithm is, respectively, up to 65% (for 12 CAVs and in average 52% for all number of crossing CAVs), and 54% less than the case where CAVs are controlled by the reservation-based approach in [1] and the lane-free method in [2]. This is, of course, in cost of higher energy consumption (due to acceleration), as the objective function of the proposed algorithm only considers minimisation of travelling time. The next subsection provides a detail analysis on energy consumption due to acceleration of different approaches, and shows that the proposed algorithm can still achieve significant improvement in crossing time while consuming the same amount of energy (due to acceleration) as the reservation-based method in [1].

Table 3.2: Performance of the proposed lane-free method for test scenario one as compared to the reservation-based method in [1] and the lane-free method in [2].

Number of CAVs →	**2**	**4**	**6**	**8**	**10**	**12**
The proposed algorithm						
Crossing time (s)	4.56	4.57	4.57	4.57	4.57	4.57
Average speed (m/s)	15.2	15.8	15.6	15.1	15.4	15.5
Standard deviation of speed	3.46	3.77	3.66	3.43	3.58	3.66
Energy consumption (kWh)	0.1	0.23	0.33	0.39	0.52	0.65
Travelled distance (m)	130	270	400	518	658	799
Reservation-based [1]						
Crossing time (s)	6.29	6.29	12.50	12.93	12.92	12.92
Average speed (m/s)	12.7	13.5	11.3	9.9	10.6	11.1
Standard deviation of speed	2.32	2.50	4.44	4.44	4.56	4.56
Energy consumption (kWh)	0.03	0.07	0.05	0.04	0.06	0.07
Travelled distance (m)	134	265	406	507	630	750
Lane-free [2]						
Crossing time (s)	10	10	10	10	10	10
Average speed (m/s)	10.1	10.0	10.0	10.0	10.1	10.0
Standard deviation of speed	0.14	0.12	0.14	0.17	0.23	0.21
Energy consumption (kWh)	0.001	0.001	0.001	0.003	0.006	0.006
Travelled distance (m)	195	387	577	773	973	1160

It is also shown in Table 3.2 that, unlike the reservation-based strategy, the resulting crossing time of the proposed algorithm does not change regardless of number of crossing CAVs. There is a similar trend for the average and standard deviation of speed of CAVs when they are controlled by the developed strategy. Also, it is evident from Table 3.2 that the standard deviation of the speed of crossing CAVs when they are controlled by the proposed algorithm is mostly less than the case when they are controlled by the reservation based strategy in [1]. Apparently, the smaller value of standard deviation of speed indicates a less diverge set of speeds (i.e., smoother travel) for the crossing CAVs.

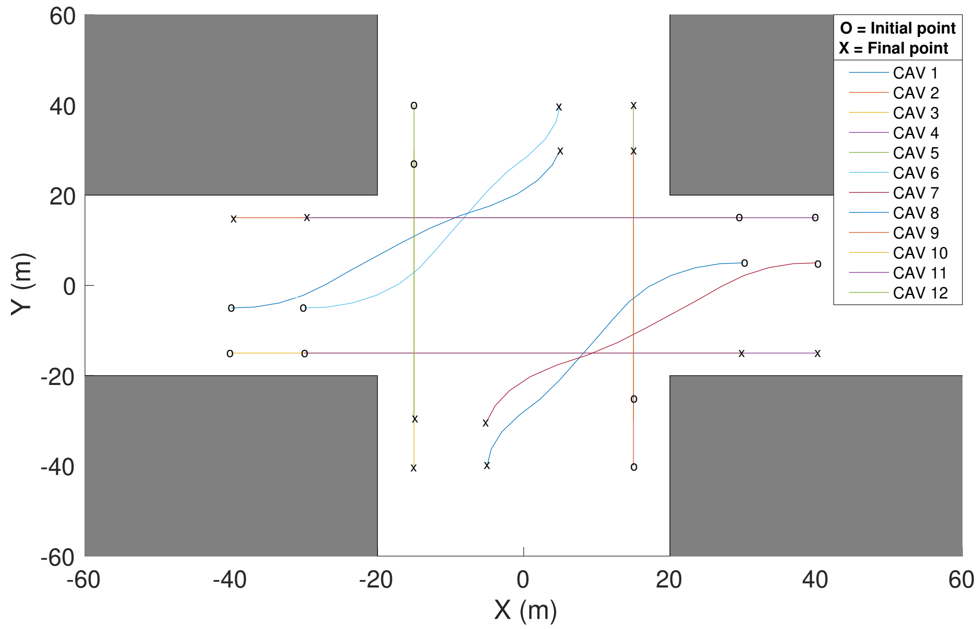
Table 3.3: Simulation results of the proposed algorithm in test scenario two, three and average of 30 scenarios for different number of CAVs.

Number of CAVs →	**3**	**6**	**9**	**12**	**15**	**18**	**21**
Test scenario two							
Crossing time (s)	4.57	4.57	4.57	4.56	4.57	4.58	4.57
Average speed (m/s)	13.2	14.5	14.5	14.2	14.1	13.8	13.6
Standard deviation of speed	3.9	4.0	3.8	3.9	4.1	4.2	4.1
Energy consumption due to acceleration (kWh)	0.11	0.27	0.4	0.5	0.64	0.7	0.8
Travelled distance (m)	170	373	561	730	910	1067	1224
Test scenario three							
Crossing time (s)	4.44	4.55	4.55	4.57	4.55	4.55	4.57
Average speed (m/s)	13.1	13.4	13.4	13.0	13.6	13.0	13.0
Standard deviation of speed	2.4	2.9	2.3	2.8	3.1	3.3	3.4
Energy consumption due to acceleration (kWh)	0.08	0.18	0.26	0.34	0.50	0.55	0.65
Travelled distance (m)	165	344	516	675	874	1002	1171
Average of 30 scenarios							
Crossing time (s)	4.51	4.56	4.54	4.56	4.57	4.58	4.58
Standard deviation of crossing time	0.06	0.02	0.09	0.06	0.08	0.10	0.10
Average speed (m/s)	13.8	13.8	13.7	13.6	13.6	13.6	13.3
Standard deviation of speed	3.3	3.6	3.2	3.5	3.6	3.6	3.5
Energy consumption due to acceleration (kWh)	0.11	0.23	0.34	0.43	0.59	0.62	0.72
Travelled distance (m)	175	361	542	701	895	1046	1206

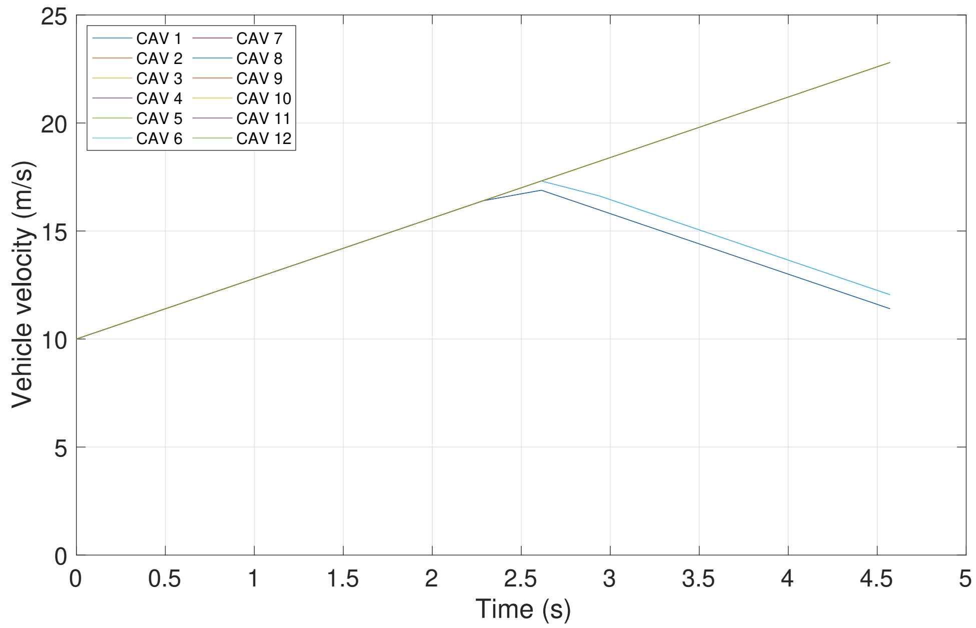
Table 3.3 shows crossing time, average and standard deviation of speed and travelled distance for different number of CAVs when they are controlled by the proposed strategy in test scenario two, three and average of 30 scenarios. As shown in Table 3.3, crossing time of CAVs in all scenarios is the same as the one in test scenario one, and again does not change regardless of the number of CAVs. This determines that the crossing time of lane-free intersections is not sensitive to the type of scenario and number of CAVs. This is an interesting outcome that shows in lane-free intersections, the crossing time of CAVs is limited by the layout of the junction rather than by the number of passing CAVs, as in traditional signalised intersections.

In fact, crossing time of CAVs cannot be theoretically smaller than the travelling time of the CAV that drives the longest distance with its maximum permissible acceleration. In other words, the minimum crossing time is dominated by the CAV that is furthest from the intersection and those closer CAVs to the junction do not change the crossing time. In all three scenarios, the initial speed of CAVs $V(t_0)$ is 10 (m/s), the maximum travelling distance (e.g, the travelling distance of the CAV that is furthest away from the intersection) $\triangle x$ is 70 (m) and the maximum acceleration $\bar{a}$ is 3 (m/s^2), and hence the theoretical lower bound of crossing time is 4.27 (s) calculated by the Newton's law $\triangle x = \frac{1}{2}\bar{a} \times t^2 + V(t_0) \times t$. The results in Table 3.2 and 3.3 show that the proposed algorithm finds a very close value to this theoretical boundary regardless of type of scenario and number of crossing CAVs. In fact, the resulting crossing time can be as close as desired to the theoretical bound in cost of deviation of final point from the desired destination point.

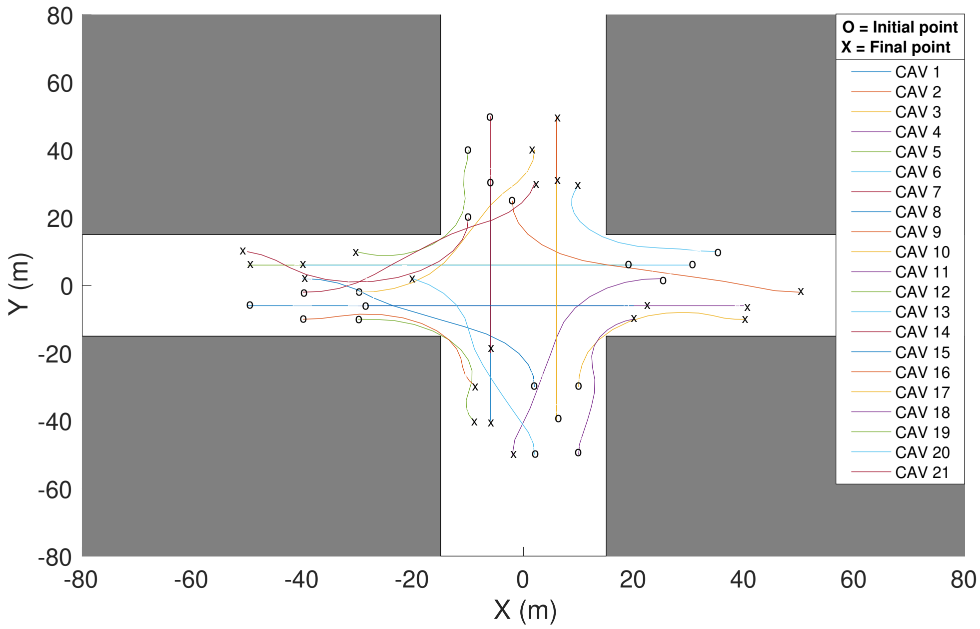
Fig. 3.2 shows the calculated optimal vehicles' motion and speed trajectory for the proposed algorithm in all three test scenarios with the maximum number of CAVs (i.e., 12 for test scenario one and 21 for the test scenario two and three). As shown in Fig.s 3.2a, 3.2d and 3.2f the proposed strategy increases and decreases the speed of CAVs linearly to avoid collisions. The slope of variation (i.e., acceleration and deceleration) is 3 (m/s^2) indicating that it is a bang-bang strategy. Moreover, the motion trajectory Fig.s 3.2b, 3.2c and 3.2e illustrate that CAVs move and use opposite lanes freely while avoiding road boundaries. The results are also visualised by a provided video on https://www.youtube.com/watch?v=S2GiGPQAfow .


O = Initial point
X = Final point
CAV 1
CAV 2
CAV 3
CAV 4
CAV 5
CAV 6
CAV 7
CAV 8
CAV 9
CAV 10
CAV 11
CAV 12
Y (m)
X (m)

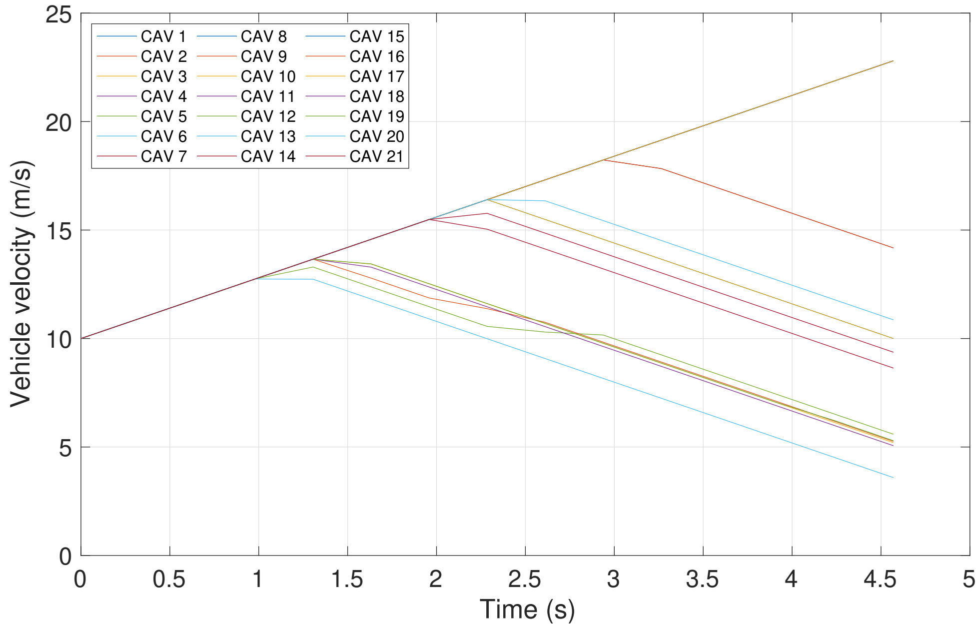
(a)


CAV 1
CAV 2
CAV 3
CAV 4
CAV 5
CAV 6
CAV 7
CAV 8
CAV 9
CAV 10
CAV 11
CAV 12
Vehicle velocity (m/s)
Time (s)

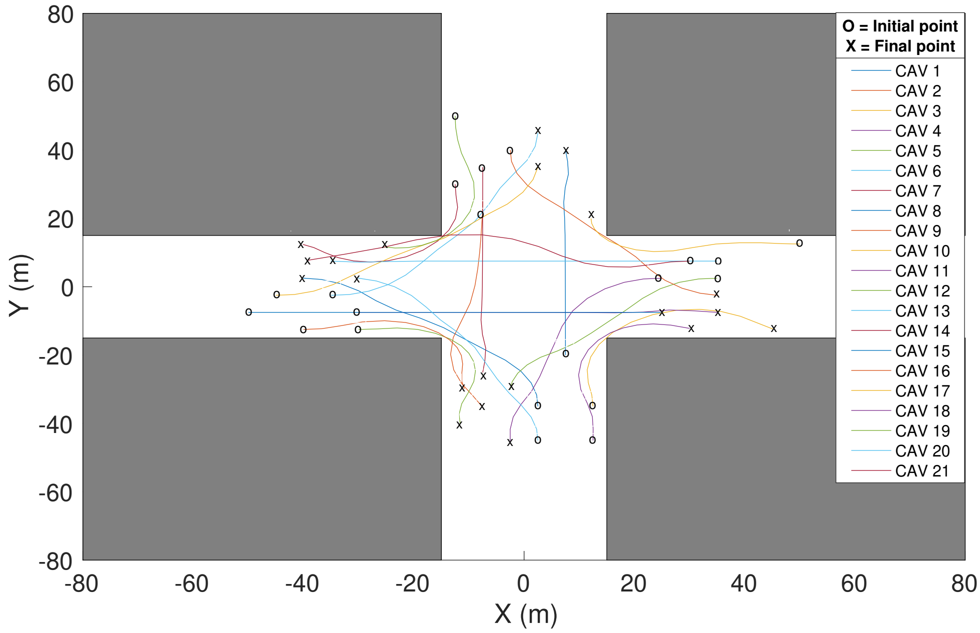
(b)


O = Initial point
X = Final point
CAV 1
CAV 2
CAV 3
CAV 4
CAV 5
CAV 6
CAV 7
CAV 8
CAV 9
CAV 10
CAV 11
CAV 12
CAV 13
CAV 14
CAV 15
CAV 16
CAV 17
CAV 18
CAV 19
CAV 20
CAV 21
Y (m)
X (m)

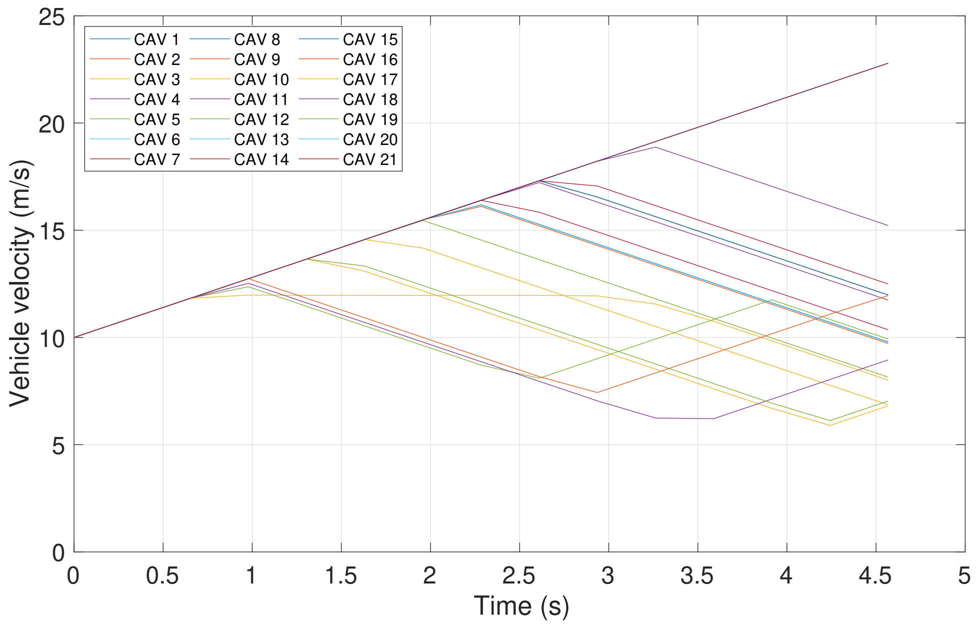
(c)


CAV 1
CAV 2
CAV 3
CAV 4
CAV 5
CAV 6
CAV 7
CAV 8
CAV 9
CAV 10
CAV 11
CAV 12
CAV 13
CAV 14
CAV 15
CAV 16
CAV 17
CAV 18
CAV 19
CAV 20
CAV 21
Vehicle velocity (m/s)
Time (s)

(d)



(e)



(f)

Figure 3.2: The calculated optimal trajectories of speed using the proposed algorithm in test scenario one, two and three for 12, 21 and 21 CAVs respectively. (a) Test scenario one's motion trajectory. (b) Test scenario one's speed trajectory. (c) Test scenario two's motion trajectory. (d) Test scenario two's speed trajectory. (e) Test scenario three's motion trajectory. (f) Test scenario three's speed trajectory.

Fig. 3.2 shows the calculated optimal vehicle motion and speed trajectory for the proposed algorithm in all three test scenarios with the maximum number of CAVs (i.e., 12 for test scenario one and 21 for the test scenario two and three). As shown in Fig.s 3.2a, 3.2d and 3.2f the proposed strategy increases and decreases the speed of CAVs linearly to avoid collisions. The slope of variation (i.e., acceleration and deceleration) is 3 (m/s^2) indicating that it is a bang-bang strategy. Moreover, the motion trajectory Fig.s 3.2b, 3.2c and 3.2e illustrate that CAVs move and use opposite lanes freely while avoiding road boundaries.

It must be noted that the possibility of collision between two consecutive control intervals is zero because the chosen sampling time is 0.152 (s) which is less than the threshold. The threshold value is calculated based on the minimum sampling time required for a CAV to travel a distance of one length (2.6 m) + one width (1.56 m) of a vehicle with its maximum permissible speed $\frac{2.6+1.56}{25} = 0.166$ (s). In other words, for the obtained minimum crossing time of 4.57 s, mid-point collisions are infeasible for any number of control intervals N_p greater than or equal to $\frac{4.57}{0.166} \approx 28$. This paper chooses a value of 30 for the number of control intervals which exceeds the threshold with a low computational time.

3.7.2 Effectiveness in terms of Energy Consumption

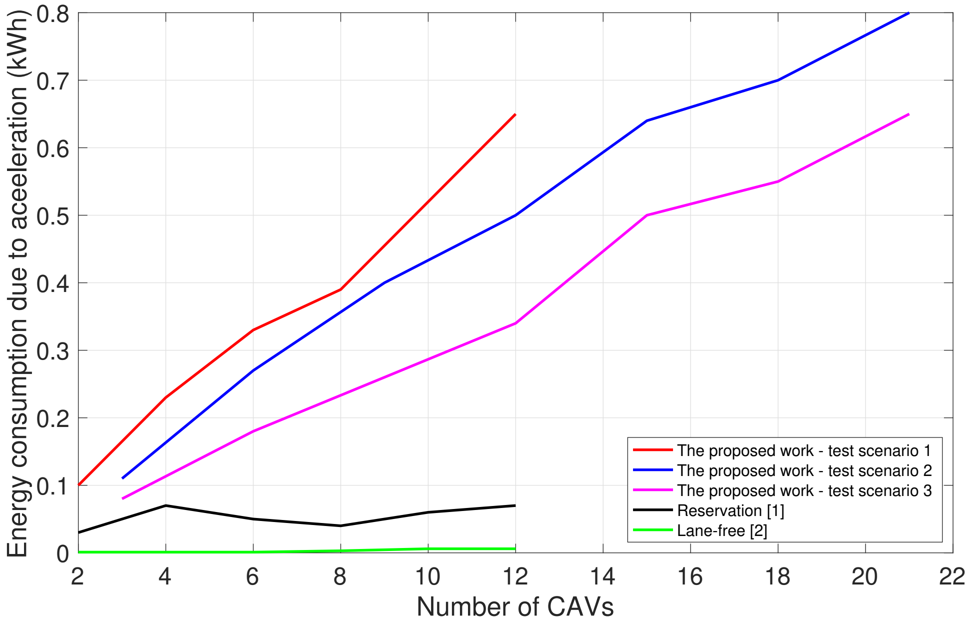
Fig 3.3a illustrates the total energy consumption due to acceleration of CAVs when the vehicles are controlled by the proposed and benchmark strategies in test scenario one. The figure also shows the total energy consumption of CAVs being controlled by the proposed strategy in test scenario two and three. The depicted graphs only consider the energy consumption due to acceleration which is calculated as follows:

$$E_i = m \int_{t_0}^{t_f} a_i(t) v_i(t) dt$$

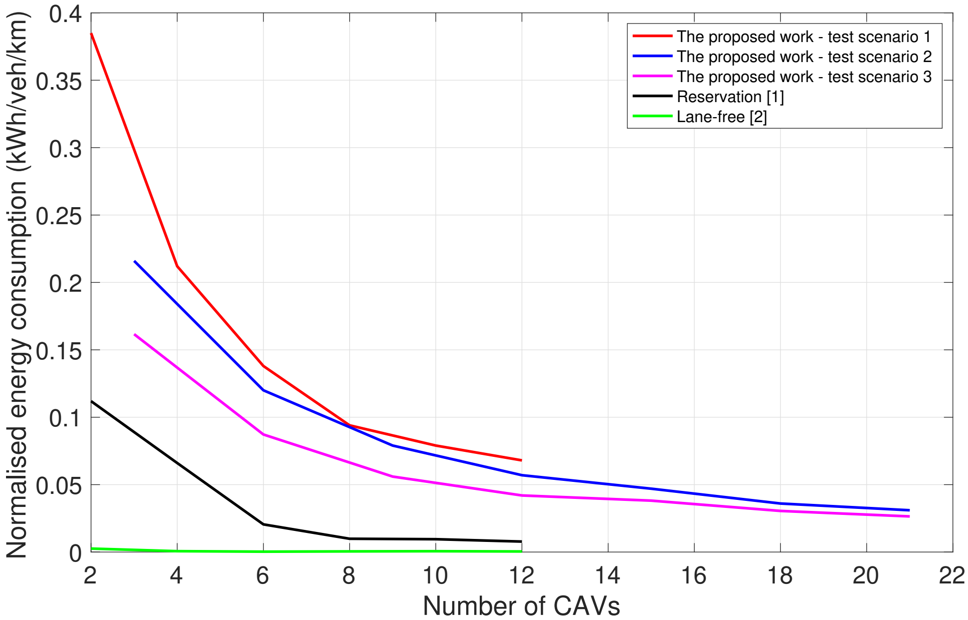
where E_i is the energy consumed by each CAV$_i$.

As seen in Fig. 3.3a, the lane-free method proposed in [2] consumes the least energy, in cost of fixing the crossing time to an unnecessarily large value (i.e., 10 s). The proposed algorithm in this paper, in contrast, consumes more energy than both the benchmark strategies because it is optimised for minimisation of crossing time, as explained in section 3.7.1.

Moreover, the resulting energy consumption (due to acceleration) of the proposed strategy linearly increases with respect to the number of crossing CAVs in all the three test scenarios. It can also be observed that CAVs consume more energy in test scenario one



(a)



(b)

Figure 3.3: Comparison of a) total energy consumption due to acceleration b) energy due to acceleration consumed per vehicle per kilometer for different number of vehicles.

than in test scenario two and three because of a longer travelling distance due to diversity of destination of CAVs. The travelled distance values of test scenario one is provided in Table 3.2 and for test scenario two and three in Table 3.3.

Fig. 3.3b, on the other hand, compares the algorithms in terms of the energy consumption by each vehicle when travels one kilometer. As seen, all the strategies tend to consume less energy per vehicle per kilometer with an increase in the number of CAVs. This is due to the fact that the number of obstacles drops by reducing the number of crossing CAVs, and hence vehicles can accelerate and pass through faster in test scenario one.

Table 3.4: Performance of the proposed lane-free method in test scenario one when the energy consumption is the same as the reservation-based strategy in [1].

Number of CAVs →	**2**	**4**	**6**	**8**	**10**	**12**
Crossing time (s)	5.23	5.30	5.97	6.18	6.24	6.27
Average speed (m/s)	13.25	13.58	11.95	11.26	11.34	11.39
Standard deviation of speeds	1.74	1.74	1.14	1.00	0.99	0.97
Energy consumption (kWh)	0.03	0.07	0.05	0.04	0.06	0.07
Travelled distance (m)	131	270	401	521	661	801

To nullify energy consumption as one of the objectives, and only compare the crossing time of CAVs when they are controlled by the proposed strategy and the reservation-based one in [1], the acceleration gain γ in (3.9) is tuned such that energy consumption due to acceleration of CAVs in both cases becomes the same. Table 3.4 shows the resulting performance of the proposed algorithm. As compared to the results of the reservation-based method in Table 3.2, the proposed algorithm reduces the crossing time up to 52% (average of 40%) when consumes the same amount of energy. Moreover, whilst the average speed of CAVs is almost similar for both the strategies, the standard deviation of speed of CAVs controlled by the proposed algorithm is much lower. This indicates that CAVs controlled by the proposed algorithm travel with similar speed, whilst some of the CAVs being controlled by the reservation-based method travel with a much higher or lower speed than the others.

Fig. 3.4 depicts that the proposed algorithm finds the Pareto front of all the crossing solutions of 12 CAVs for different values of acceleration gain γ. As seen, the proposed strategy can achieve shorter crossing time than both the reservation-based method [1] and lane-free method [2] while consuming the same amount of energy.

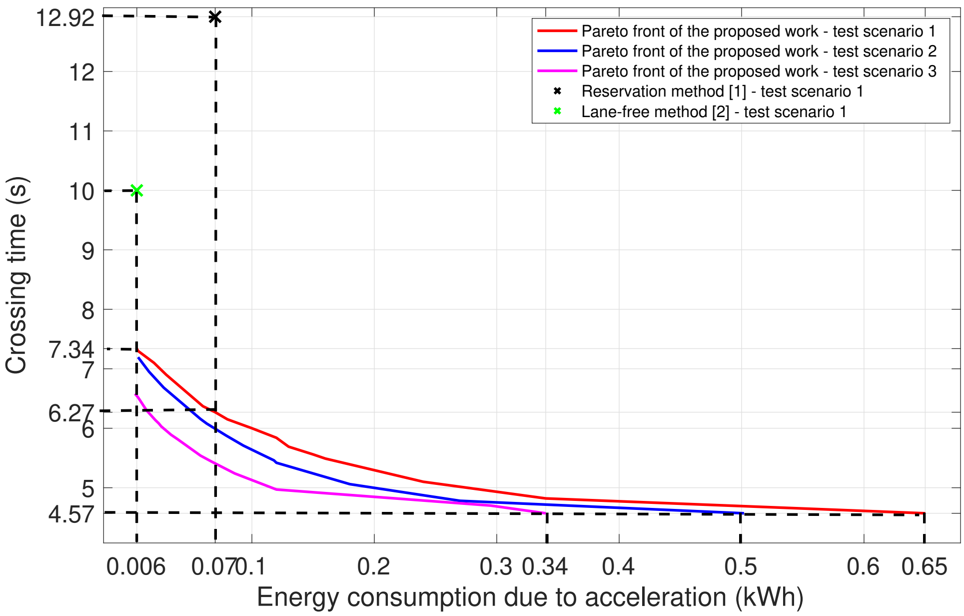


Figure 3.4: Energy vs crossing time (Pareto front) of 12 CAVs controlled by the proposed strategy as compared to the results by the reservation method in [1] and the lane-free method in [2].

Furthermore, Fig. 3.4 shows that the minimum crossing time of the proposed strategy in all three scenarios is 4.57 s. This indicates that the minimum crossing time of the proposed algorithm is independent of the type of scenario and confirms the data provided in Tables 3.2 and 3.3. However, it can be seen from Fig. 3.4 that the crossing time is slightly dependent to the type of scenario when CAVs consume minimal energy. This can be due to CAVs finding trajectories that tend to be energy efficient but are longer to travel.

It is worth noting that the minimum crossing time of CAVs can be as close as possible to its theoretical lower bound however at the cost of deviation from the final point.

3.7.3 Effectiveness in terms of Passenger Comfort

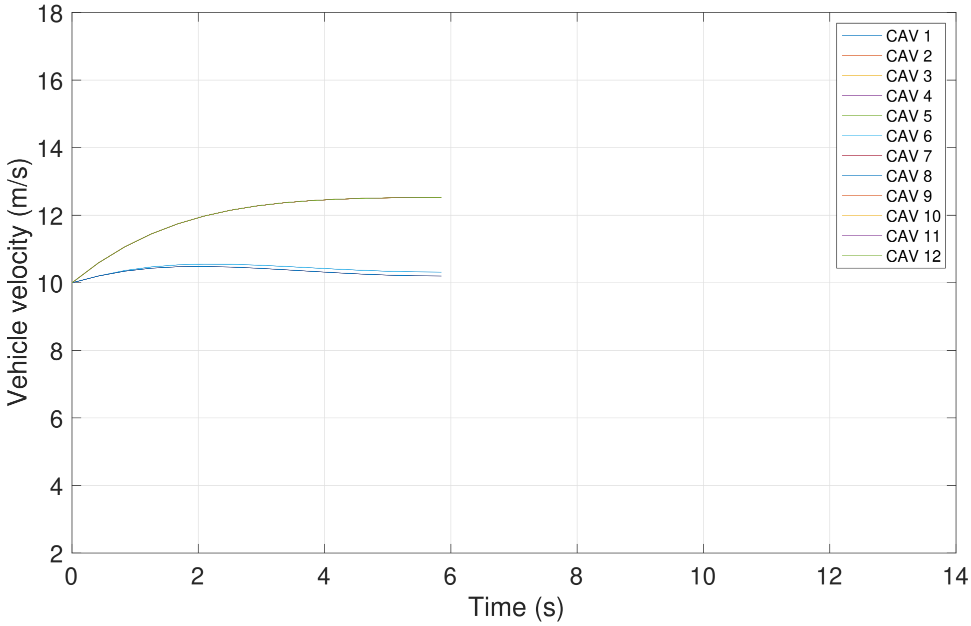
Fig. 3.5 compares the calculated optimal vehicle speed, longitudinal and lateral acceleration (or deceleration) trajectories by the proposed algorithm with the results of the reservation-based method in [1] for 12 CAVs in test scenario one, when the energy consumption due to acceleration is the same ($\beta << 1$).

Fig. 3.5a as compared to Fig. 3.5b shows that the vehicles travel within a much narrower range of speed and hence the passengers experience similar feeling of speed when CAVs are controlled by the proposed algorithm as opposed to the reservation-based method in [1].

Moreover, as shown in Fig. 3.5c, the maximum deceleration of CAVs when they are controlled by the proposed strategy is 1.4 (m/s^2) which is much less than the maximum permissible value of 3 (m/s^2). The acceleration of all CAVs also converges to zero at their destinations. Fig. 3.5d, on the other hand, shows that some of the CAVs controlled by the reservation-based algorithm in [1] decelerate with the maximum permissible value, which is not converged to zero.

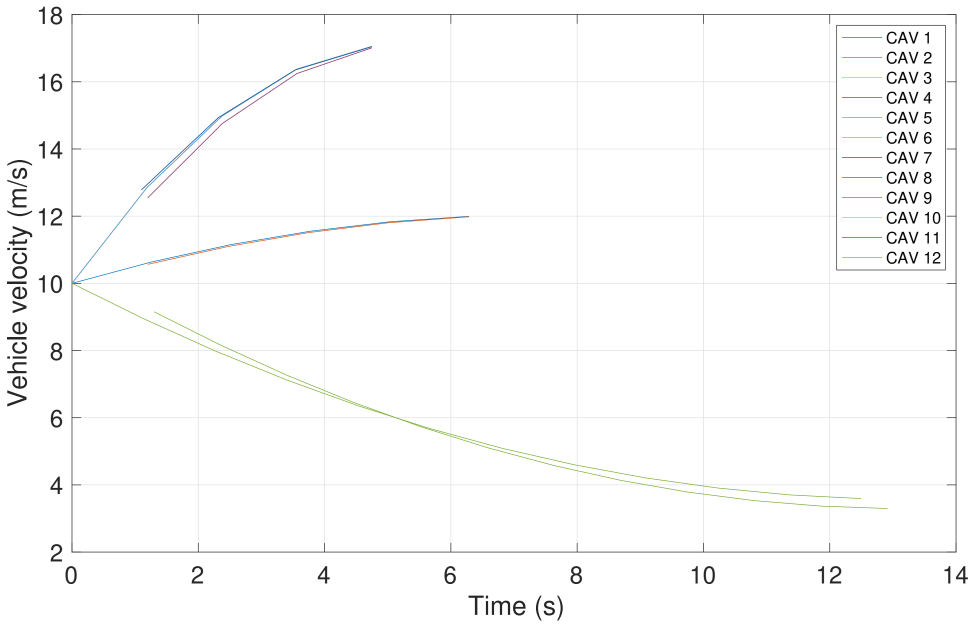
The maximum jerk of both algorithms is around 0.6 m/s^3, however, whilst jerks of CAVs controlled by the proposed strategy converge to zero, the passengers feel an uncomfortably constant jerk during the crossing time when CAVs are controlled by the reservation-based method in [1].

In contrast, Fig. 3.5e and 3.5f illustrate that passengers of the cornering vehicles, i.e., CAV$_i$ $i \in \{1, 6, 7, 8\}$, experience sharper variation of lateral acceleration when the vehicles are controlled by the proposed algorithm than the algorithm in [1], even though the maximum values are almost similar. This is due to the fact that the proposed algorithm generates higher lateral acceleration to achieve shorter travelling time.



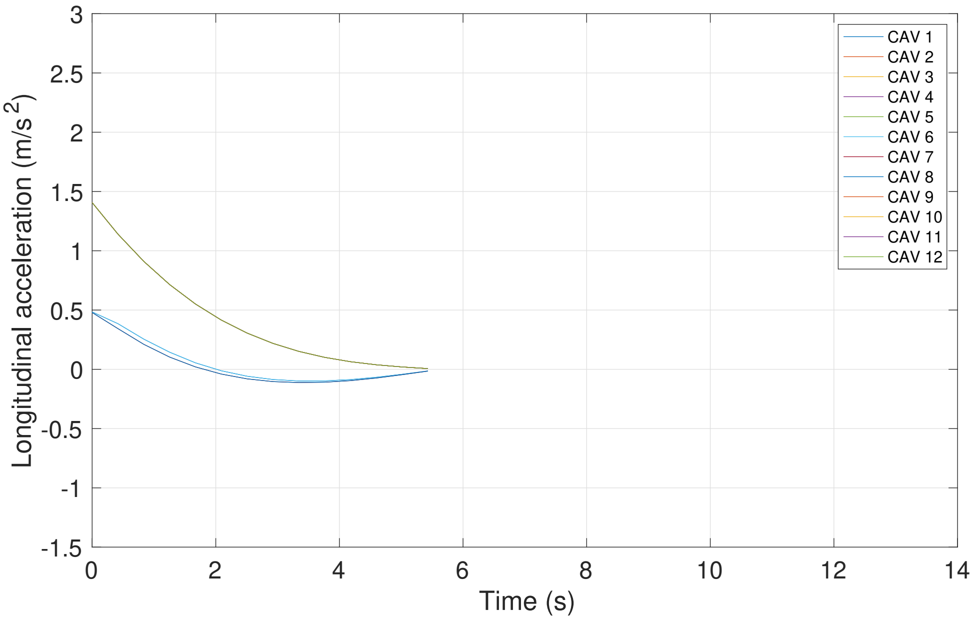
Vehicle velocity (m/s)
Time (s)
CAV 1
CAV 2
CAV 3
CAV 4
CAV 5
CAV 6
CAV 7
CAV 8
CAV 9
CAV 10
CAV 11
CAV 12

(a)

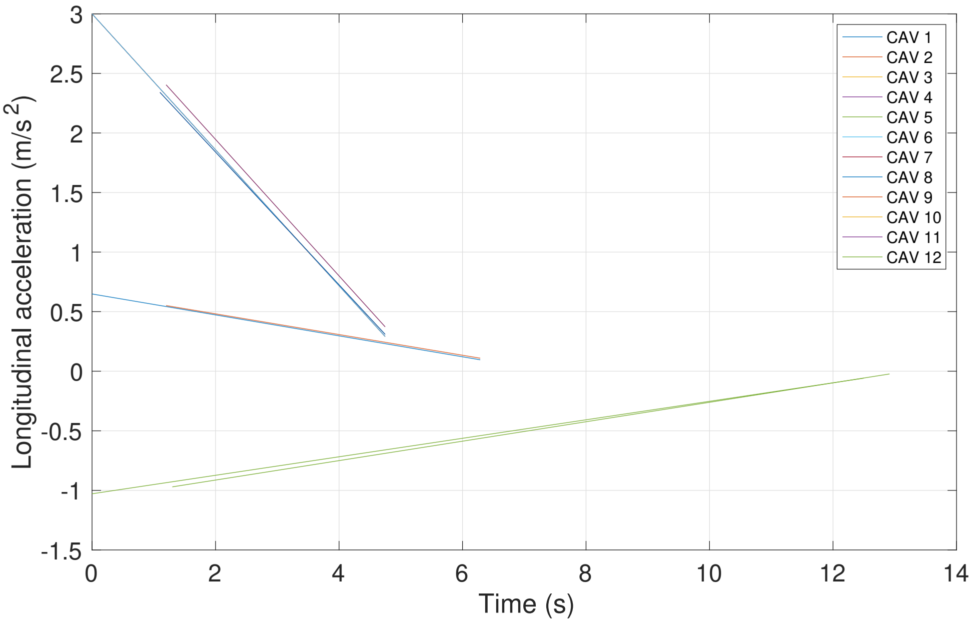


Vehicle velocity (m/s)
Time (s)
CAV 1
CAV 2
CAV 3
CAV 4
CAV 5
CAV 6
CAV 7
CAV 8
CAV 9
CAV 10
CAV 11
CAV 12

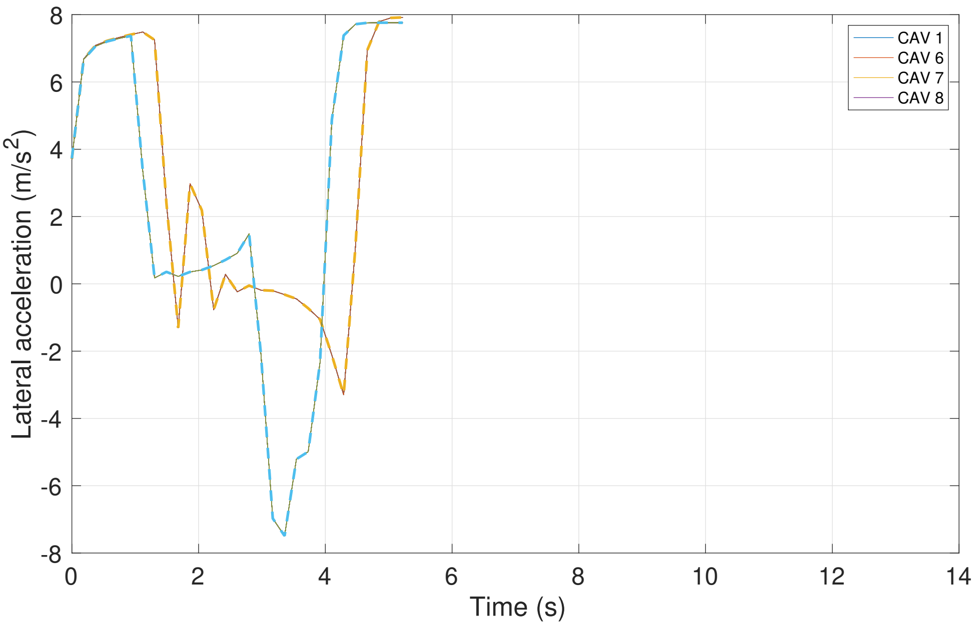
(b)


Longitudinal acceleration (m/s^2)
Time (s)
CAV 1
CAV 2
CAV 3
CAV 4
CAV 5
CAV 6
CAV 7
CAV 8
CAV 9
CAV 10
CAV 11
CAV 12

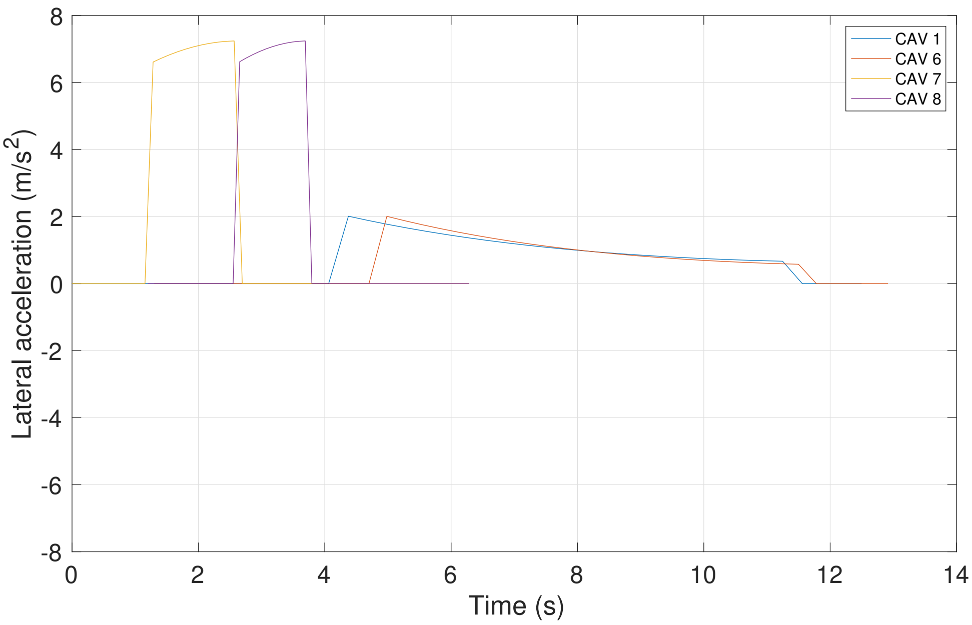
(c)


Longitudinal acceleration (m/s^2)
Time (s)
CAV 1
CAV 2
CAV 3
CAV 4
CAV 5
CAV 6
CAV 7
CAV 8
CAV 9
CAV 10
CAV 11
CAV 12

(d)



(e)



(f)

Figure 3.5: The calculated optimal trajectories of speed and acceleration using the proposed algorithm and reservation-based method [1] in test scenario one for 12 CAVs, when energy consumption is the same. (a)The proposed algorithm's speed trajectory. (b)The reservation-based method [1] speed trajectory. (c)The proposed algorithm's longitudinal acceleration trajectory. (d)The reservation-based method [1] longitudinal acceleration trajectory. (e)The proposed algorithm's lateral acceleration trajectory. (f)The reservation-based method [1] lateral acceleration trajectory.

3.7.4 Computational Time and Implementation Considerations

Fig. 3.6 depicts computational time of the proposed algorithm for different number of CAVs in test scenario two. The computation time of each number of CAV is the average of 10 times of running the scenario. The standard deviations of all the tests are less

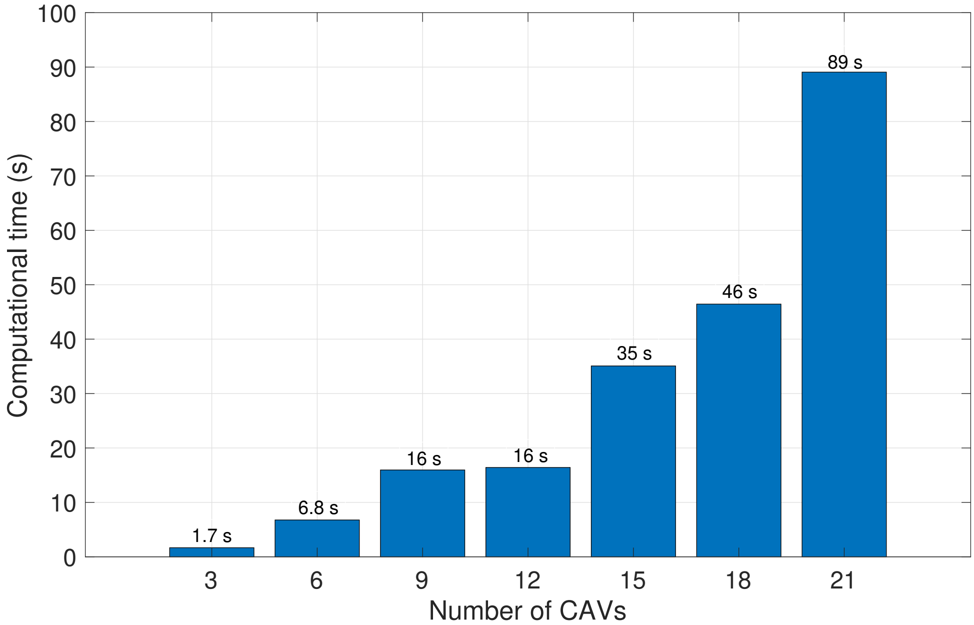


Figure 3.6: The average computational time of 10 runs of the proposed strategy for different number of CAVs with test scenario two. The standard deviation of the 10 runs for each number of CAVs is less than 0.5%.

than 0.5% which is negligible and are not shown. Fig. 3.6 shows that the computational complexity of the proposed algorithm is of the order of $O(e^{0.13N})$ in terms of number of CAVs N.

The real-time implementation of the proposed strategy will be in an MPC framework. MPC can solve the proposed OCP with a shorter sampling time and compensate the resulting errors by employing an additional feedback mechanism over a receding horizon. The results of the study shows the sampling time as $\frac{4.57}{30} = 0.152\ s$. However, in order to cover all the dynamics of the vehicle, three times shorter than the current sampling time is required which will increase the computational time significantly. The raised issue in the computational time can be addressed by solving the OCP in a receding horizon fashion (MPC). MPC can take also into account the uncertainties of the environment by employing the feedback mechanism that constantly adjusts the control inputs based on the current state and any environmental disturbances. In addition, the available techniques for real-time implementation of MPC, including explicit MPC makes MPC a potential solution for real-time implementation. For more information on the formulation of MPC the reader is referred to Section 2.4.1.

Moreover, the proposed strategy will be implemented in a decentralised frame. As explained in Section 2.4.1, solving the resulting MPC framework of OCP (3.8) within a centralised framework presents challenges due to communication uncertainties that arise from the fact that CAVs may not always be connected. Therefore, the MPC can be

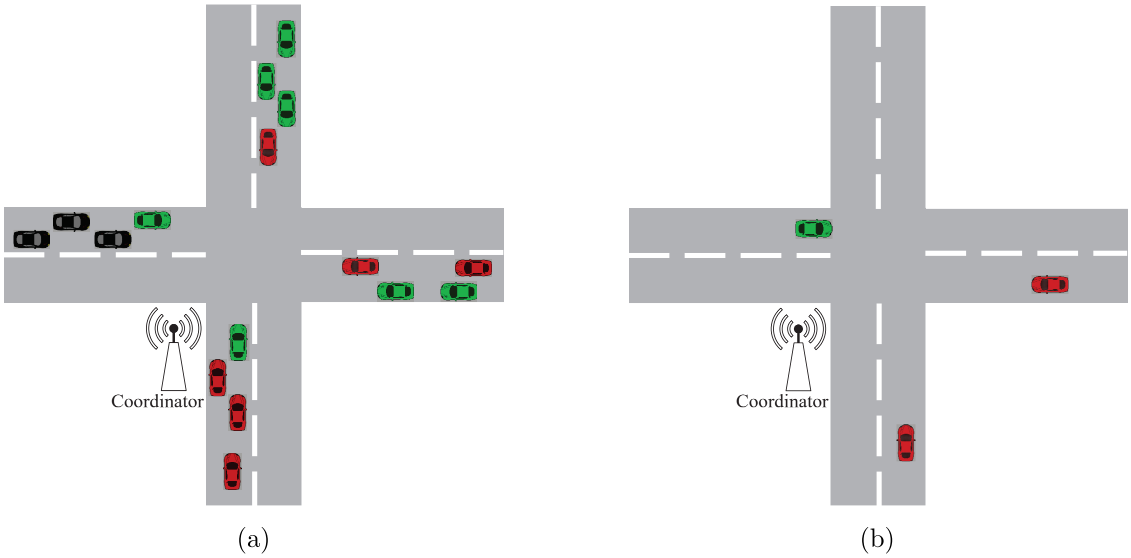


Figure 3.7: The proposed algorithm calculates trajectories for the CAVs when a)the number of vehicles reaches the practical limit; b)a vehicle reaches the beginning of the intersection.

implemented in a decentralised framework where each CAV computes its own trajectory by only communicating to CAVs in its communication range. Such a decentralised framework also may potentially improve the computation time.

The real-time implementation of the proposed algorithm will be solved for a group of vehicles within a specified control zone. The control zone in this study is defined based on the communication range of the coordinator. As an example of a real-time scenario, the number of vehicles entering the zone will be counted and when either this number reaches its practical limit (depicted in Fig. 3.7a) or one of the vehicles becomes close to the intersection (depicted in Fig. 3.7b), the OCP is solved for the vehicles within the control zone.

To enable the proposed algorithm in the real world, three types of hardware are required, with the first being an X-by-wire system for CAVs. This electronic system controls the vehicle's functions, including steering, acceleration, and braking, and is integrated with sensors and software that enable autonomous operation. Typically, the X-by-wire system in a CAV is equipped with sensors that collect data about the vehicle's surroundings and internal state, such as speed, position, and orientation. This data is then used by the vehicle's software to determine the appropriate actions, such as adjusting steering or braking to avoid obstacles or maintain a safe distance from other vehicles.

Secondly, CAVs require devices such as Global Positioning System (GPS) to accurately find their positions on the Earth's surface. Unlike GPS that provides a position with an accuracy of around 5-10 meters, Differential Global Positioning System (DGPS) can improve the accuracy to within a few centimeters. DGPS achieves this improved accuracy by comparing GPS signals received by a reference station with those received by a moving receiver on the CAV, and applying a differential correction.

Lastly, a coordinator device is required to be placed at intersections to receive the information of CAVs such as position, speed and destinations. This device communicates with the CAVs within its range and calculates a collision-free trajectory for all the surrounding CAVs and send the obtained trajectory to each one of them. Then each CAV calculates its own required control inputs based on the received trajectory. The controls include the steering angle and acceleration/deceleration which are sent to the vehicles actuators to follow the received trajectory and pass through the intersection.

3.8 Summary

This chapter formulated and solved the lane-free crossing of CAVs through intersections as an optimal control problem that minimises the overall crossing time and energy consumption due to acceleration of CAVs while avoiding obstacles. The proposed formulation employs dual problem theory to substitute the non-differentiable constraints of collision avoidance with the dual problem of a corresponding sufficient condition.

The resulting smoothed OCP is then solved by CasADi to generate a trajectory for safely cross of multiple CAVs through a junction within the minimum time. It is shown that the lane-free crossing is capable of significantly reducing the crossing time as compared to the state-of-the-art reservation-based strategy, whilst consuming similar energy.

The presented results show that the proposed strategy finds the minimum crossing time of CAVs which is very close to its theoretical limit. Also, it shows that the calculated time only relies on the layout of intersection and is independent of the number of crossing CAVs or their manoeuvres. This makes the results of the proposed algorithm a suitable benchmark to evaluate the performance of other control strategies of the CAVs crossing intersections.

Chapter 4

Capacity Analysis of Lane-Free Intersections

Lane-free crossing of CAVs through intersections allows them to utilise the most of spatial-temporal area of intersection which increases the capacity of junctions. This chapter provides a framework that evaluates the capacity of intersections when CAVs are crossing in a lane-free order. The framework involves a measure and an algorithm to calculate the capacity of lane-free intersections. The capacity of a lane-free intersection is compared against the capacity of the intersection when signalised and calculated based on highway capacity manual. In addition, two adaptive traffic controllers namely max-pressure and Webster are developed to measure the maximum throughput of a signalised intersection for the same scenario as the lane-free intersection. The chapter also includes a sensitivity analysis of crossing time and capacity for lane-free intersections with respect to variation in initial speed, maximum permissible speed and acceleration.

4.1 The Framework to Quantify Capacity of the Lane-Free Intersections

Conventionally, capacity of intersections (both the signalised and unsignalised) are measured using a set of collected data from either real-time observation of vehicles [78] or running a micro-simulation [125, 126]. For example, capacity of each lane of an unsignalised all-way stop-controlled (AWSC) intersection is measured by gradually increasing the flow rate of the lane in the simulator until the degree of utilisation (DoU) of the lane reaches one, which happens when throughput of the lane is equal to its capacity.

DoU represents the fraction of capacity being used by vehicles and is defined as follows

[78]:

$$x = \frac{vh_d(x)}{3600} \tag{4.1}$$

where x denotes the degree of utilisation, v refers to flow rate (throughput) (veh/h) of the lane and h_d is the departure headway (s) that is a function of x and is calculated as a stochastically weighted average of the saturation headway of all combinations of possible degrees of conflict and number of crossing vehicles. The highway capacity manual [78] proposes an iterative algorithm to calculate the value of x and h_d for any given v based on the identified values from the available large set of real data.

However, such real-time data are not available for lane-free crossing of CAVs because of the lack of real infrastructure or realistic simulators that consider the collaborative behaviour of enough number of heterogeneous CAVs crossing an intersection. The remaining of this section introduces a new measure and a calculating algorithm of the capacity of lane-free intersections.

4.1.1 The proposed measure of the capacity

Intersections can host limited number of vehicles at the same time and if the intersection capacity exceeds the waiting time of crossing vehicles will significantly increase. Therefore, to evaluate the capacity of intersections a suitable measure must consider the maximum number of vehicles and the time that it takes for those vehicles to pass through the intersection. In effect, the following measure is proposed to calculate the capacity of the lane-free intersections:

$$C = max\{\frac{3600 \times N_v}{T}\} \tag{4.2}$$

where C is the capacity (veh/h) of the intersection, N_v denotes the number of crossing CAVs (veh) and T represents the time (s) that takes for those vehicles to fully cross the intersection.

Equation (4.2) requires a simulator to gradually increase the number of vehicles N and measuring their minimum crossing time T_{min} to calculate the throughput $3600 \times N_v/T_{min}$ until the throughput starts dropping. The last value of the throughput just before dropping is the capacity of the intersection.

The next sections present methods to find N_v and T for the lane-free and signalised intersections.

Table 4.1: Main parameters of the proposed algorithms and their values

Parameter(s)	**Unit**	**Value(s)**
Maximum speed	(m/s)	25
Maximum acceleration	(m/s^2)	3
Initial speed	(m/s)	10
Prediction horizon	($\times$ sampling times)	15
Safe margin between CAVs	(m)	0.1
Vehicle length	(m)	4.5
Vehicle width	(m)	1.8

4.1.2 The proposed algorithm to calculate the capacity

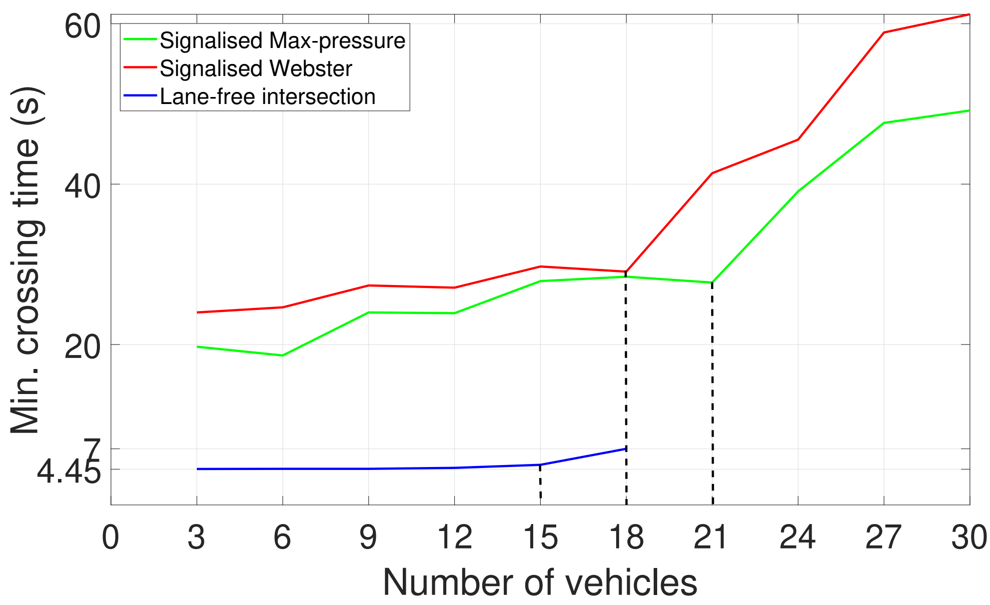
The central theme of the proposed algorithm to solve (4.2) is to use the minimum-time crossing method developed in Chapter 3 to calculate the minimum crossing time T_{min} of the lane-free intersections for a given number N_v of CAVs. To calculate the minimum time T_{min}, the OCP in (3.8) is solved as a single objective problem where the acceleration's gain in the objective function (3.9) is set to zero.

It is already shown in Section 3.7 that, unlike the signalised intersections, the crossing time of CAVs in a lane-free order is independent of the scenario (i.e., the initial positions and destinations of vehicles). Thus, (4.2) is solved for a sample scenario with a low number of crossing CAVs (e.g., three) and then new CAVs are gradually added to the scenario until the throughput reaches the capacity as explained above. This method can also be applied to calculate the maximum throughput of the signalised intersections for a given scenario.

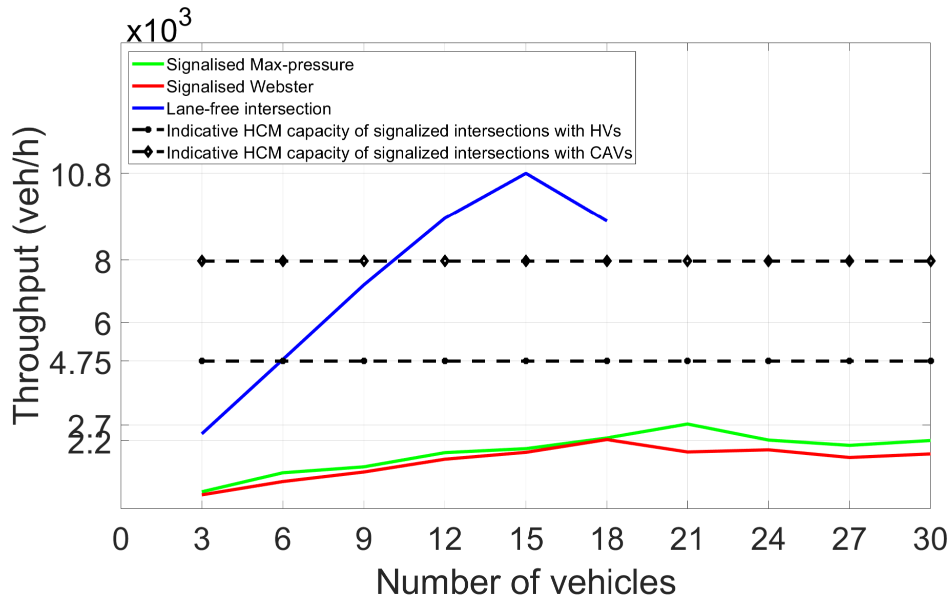
4.2 Capacity analysis of the Lane-Free Intersections

The capacity of the lane-free intersection in Fig. 2.6 is calculated in this section based on the measure (4.2) and using the introduced algorithm in Chapter (3) that computes the minimum crossing time of CAVs for a given number of vehicles. Table 4.1 summarises the critical parameters that are used throughout the calculations.

Fig. 4.1a shows the calculated minimum crossing times which are fairly constant for a wide range of the number of crossing CAVs. However, there is a sharp increase after exceeding the threshold of 15 crossing CAVs showing that the capacity is reached. Equation (4.2) is used to measure throughput of the intersection based on the calculated minimum crossing times and the peak of the calculated throughput is the capacity of the intersection. Fig. 4.1b illustrates that the capacity of the studied lane-free intersection is 10,800 $CAVs/h$ where the throughput starts dropping.



(a)



(b)

Figure 4.1: (a) The calculated crossing times by the lane-free algorithm and signalised max-pressure and Webster controllers for different number of vehicles; (b) The corresponding throughput obtained by the proposed measure as well as HCM indicative capacity of signalised intersections for both HVs and CAVs. The headway of CAVs is assumed as 1.13 s which is an average of the provided stochastic values in [6].

To compare the capacity of the lane-free intersection against signalised intersections, this study employs the HCM [78] capacity calculations for the signalised intersection with HVs. HCM defines the capacity of signalised intersections based on the saturation flow rate of each lane multiply by a green ratio f accounting for lost times due to changing phases. Considering a cycle length of 120 s and a lost time of 5 s, the green ratio is

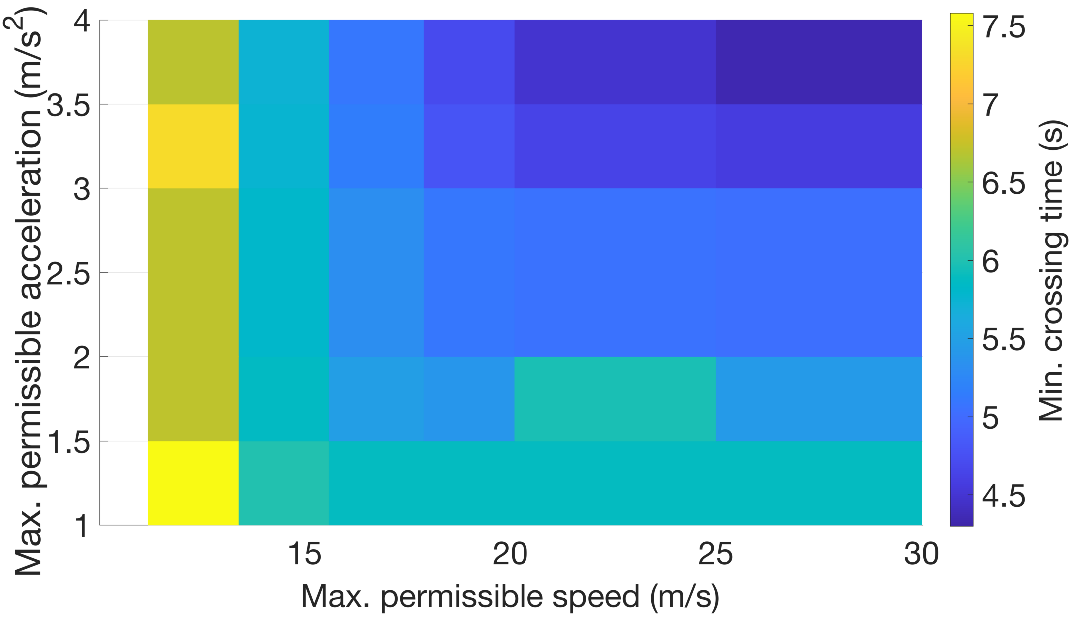
$f = \frac{120-4*5}{120} = 0.8\bar{3}$. Thus, a recommended saturation flow rate of 1900 $HVs/h/ln$ gives the capacity of the three-lane intersection in Fig. 2.6 as $1900 \times 3 \times 0.8\bar{3} = 4750$ HVs/h, which is called hereby as the HCM indicative capacity of the signalised intersection with HVs. Fig. 4.1b displays the calculated value of this indicative capacity as a horizontal line. It is worth noting that the HCM indicative capacity is independent of the number of crossing vehicles and is overlapped just for comparison. Fig. 4.1b shows that the capacity of the studied intersection when CAVs crossing in a lane-free order is 127% higher than the capacity of the same intersection when signalised and with HVs. This massive jump in capacity is due to the facts that CAVs have shorter headway, do not stop by traffic lights and, most importantly, collaborate to utilise the maximum spatial-temporal area of the intersection to minimise the crossing time.

In case of only CAVs crossing the signalised intersection, the capacity increases due to a shorter headway of CAVs than HVs. However, there is not an exact value for the headway of CAVs because this value significantly depends on the controller behaviour and hence path planning algorithms of CAVs. In this light, there is wide range of headway values provided in the literature [6, 127, 128]. The present work considers a headway of 1.13 s for CAVs which is an average of the provided stochastic values in [6]. Thus, the saturation flow rate of each lane is increased to 3,186 $CAVs/h$ and the capacity of the same signalised intersection for CAVs is calculated as 7,964 $CAVs/h$. This indicative HCM capacity of signalised intersections with CAVs is shown in Fig. 4.1b to compare with the lane-free intersection. As it can be seen, the strategy of lane-free crossing improves the capacity of the intersection by 36% as compared to signalised crossing with CAVs.

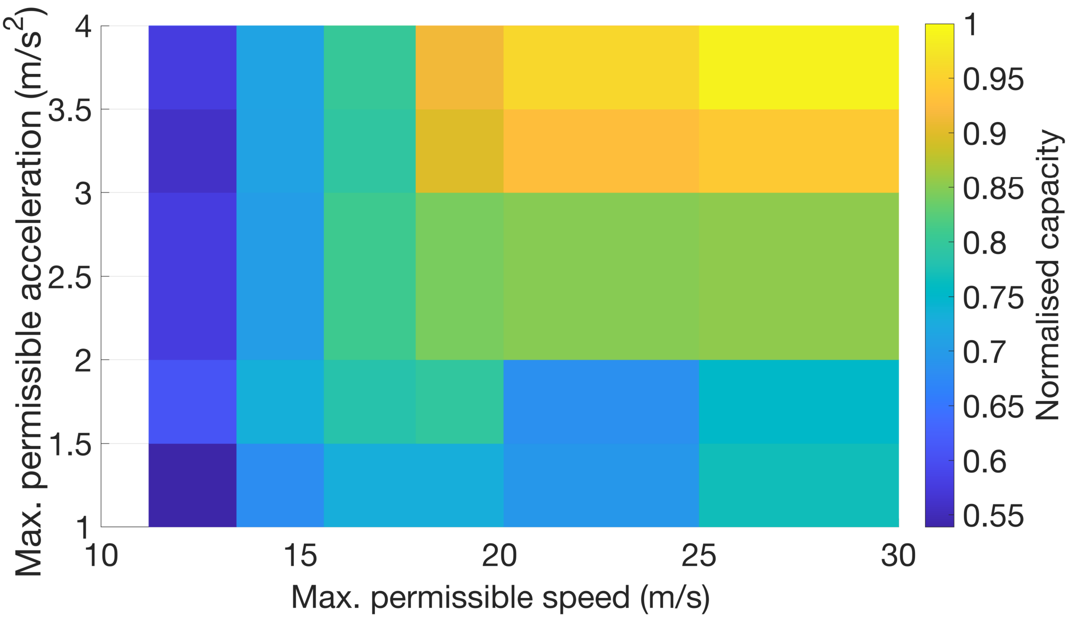
However, using the concept of the saturated flow rate to calculate the capacity of a signalised intersections with crossing CAVs seems not to be accurate because: i) there is a large discrepancy in the reported values of the CAVs' headway, ii) the previously reported headway of CAVs did not consider the collaborative and heterogeneous nature of the algorithms of CAVs, and iii) lateral dynamics of the vehicles on the truing lanes are not considered for the calculation of saturation flow rate. In fact, the provided results in this paper for the capacity of the lane-free intersection suggest that an indicative value for the CAVs' headway $T_{h,CAVs}$ can not be smaller than 0.83 s ($T_{h,CAVs} \geq \frac{3\times3600\times0.83}{10800} \geq 0.83$).

As previously mentioned, unlike the capacity of lane-free intersections, the capacity of signalised ones depends on the crossing scenario. To show this, two adaptive traffic controllers, max-pressure [129] and Webster [130] are applied to the same intersection for the same scenario as the lane-free intersection. Both the max-pressure and Webster algorithms are simulated in SUMO with the help of TraCI for gradually increasing number of HVs based on the works in [131]. Webster in [130] derived a formulation that calculates the cycle length of traffic lights. The derived cycle length is used to find the green time of each phase to allow vehicles to cross the intersection. Similarly, the max-pressure algorithm computes the signal timings, however, the green time of each phase is calculated based on the number of vehicles in the incoming and outgoing lanes [129]. Whilst Webster is a well-known algorithm for timing control of traffic lights, it is already shown that the max-pressure algorithm yields the lowest travelling time, queues length and crossing delays among all the state-of-the-art controllers [131], including the algorithms based on the self organising [132, 133], deep Q-network [134], deep deterministic policy gradient [135] and Webster methods.

Fig. 4.1a shows the SUMO simulated crossing times of different number of HVs through a signalised version of the intersection in Fig. 2.6 when the traffic lights are controlled by the max-pressure and Webster algorithms. As observed, the crossing time of max-pressure and Webster controllers increases significantly after the number of crossing vehicles exceeds the thresholds of, respectively, 21 and 18 vehicles. To calculate the corresponding maximum throughput for these adaptive controllers the measure (4.2) is employed and the results are shown in Fig. 4.1b as compared to the lane-free intersection for the same crossing scenario. From Fig. 4.1b, it can be seen that the maximum throughput of the scenario using two state-of-the-art traffic controllers are, respectively, $2,726$ (veh/h) and $2,227$ (veh/h). Hence, the capacity of the lane-free intersection is, respectively, 296% and 385% larger than the maximum throughput of max-pressure and Webster for that particular scenario. It is clear that the calculated maximum throughput values are not the same as HCM capacity of signalised intersections with HVs and this indicates that the capacity of signalised intersections is dependent to the type of scenario. This difference is due to HCM capacity calculations assuming that unlimited number of vehicles are queued in lanes, however, this might not be true in real world. Therefore, the proposed measure (4.2) and the SUMO simulator of max-pressure and Webster controllers can be employed to calculate the maximum throughput of signalised intersections for any desired scenario.



(a)



(b)

Figure 4.2: Sensitivity of the (a) crossing time and (b) capacity of the lane-free intersection in terms of the maximum speed and acceleration of CAVs. Initial speed of the vehicles is 10 (m/s)

4.3 Sensitivity Analysis of the Capacity and Crossing Time of the Lane-Free Intersections

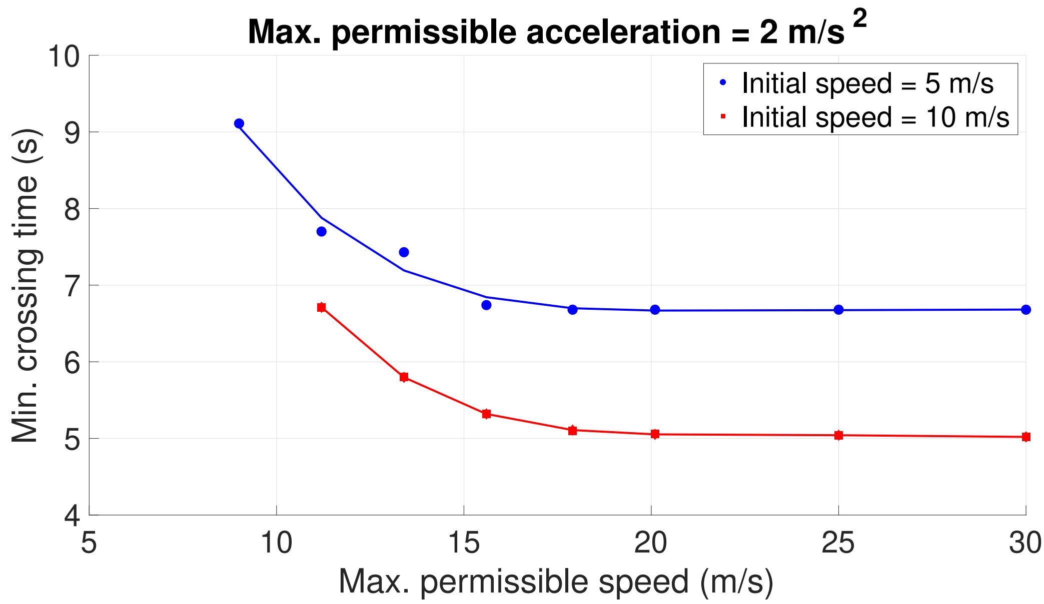
Fig. 4.2a and Fig. 4.2b, respectively, show the variation of the crossing time and the normalised capacity of the studied lane-free intersection due to changes in the maximum speed and acceleration of CAVs. As seen, the larger the maximum permissible speed and acceleration of the vehicles are, the shorter crossing time and equivalently the larger capacity are achieved. The capacity of the intersection reaches to its top value when the maximum allowed speed and acceleration are, respectively, 30 (m/s) and 4 (m/s^2). The initial speed of vehicles is 10 (m/s). Apparently and as shown in Fig. 4.2, relaxing the range of acceleration without expanding the range of speed has a very limited effects on the capacity.

Fig. 4.3 provides more detail on the results of Fig. 4.2. Samples of the results in Fig. 4.2 for two different values of the maximum permissible accelerations and speeds are illustrated separately in Fig. 4.3. As seen, the best crossing time of CAVs and equivalently the maximum capacity of the lane-free intersection improves by 28% due to an increase of the maximum acceleration from 2 (m/s^2) to 4 (m/s^2) and when CAVs enter the control area of the intersection with the initial speed of 5 (m/s). Fig. 4.3 also shows that doubling the initial speed from 5 (m/s) to 10 (m/s), the best crossing time and hence the maximum capacity increases by, respectively, 28% and 19% for the maximum accelerations of 2 (m/s^2) and 4 (m/s^2).

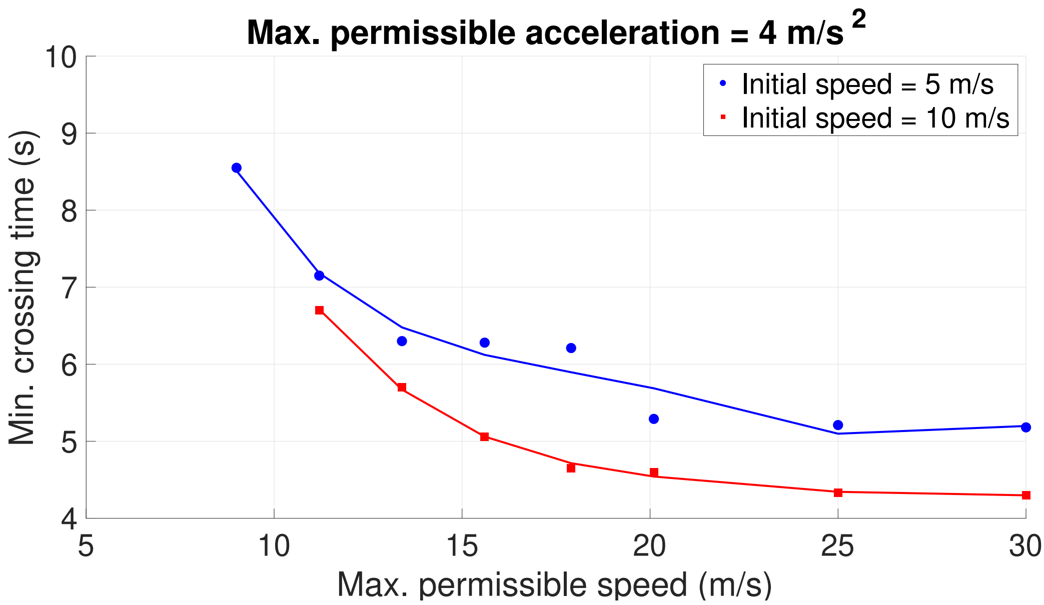
Furthermore, Fig. 4.3 shows that the maximum permissible speed also affects the minimum crossing time of CAVs and hence the capacity of the lane-free intersections to a certain limit. As shown, the capacity of the studied lane-free intersection increases logarithmically with a factor of 44% by rising the limit of the maximum permissible speed of CAVs up to around 18 (m/s) when the maximum allowable accelerations is 2 (m/s^2) (and 54% when it is 25 (m/s) at the maximum allowable acceleration fo 4 (m/s^2)). The capacity stays steady after these limits.

Fig. 4.4, on the other hand, provides a similar analysis of the minimum crossing time and equivalently maximum throughput of the intersection in Fig. 2.6 when there is a traffic light that controls the flow of intersection with the max-pressure and Webster state-of-the-art algorithms. Unlike the results in Fig. 4.3, Fig. 4.4 shows that the maximum throughput of the same intersection but when it is signalised is only slightly sensitive to the maximum permissible acceleration and does not vary by increasing the maximum allowable or initial speeds of the crossing vehicles. This is because of the fact that traffic lights oblige HVs to stop before the signalised intersections no matter what the vehicles' speed are, whilst CAVs can cross the lane-free intersections at all directions continuously and with no interrupts. The crossing time T of these stopped vehicles is dominated by the human reaction time with a mean fixed to a constant value. Hence, referring to Equation

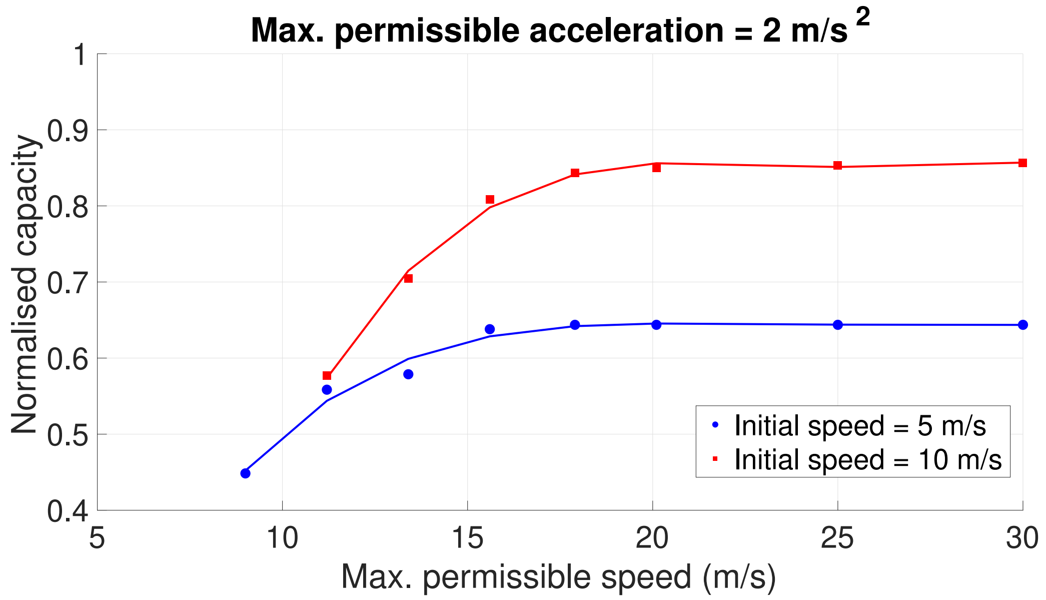
(4.2), maximum throughput of the signalised intersections is insensitive to the parameters and is dominant by the human factors.



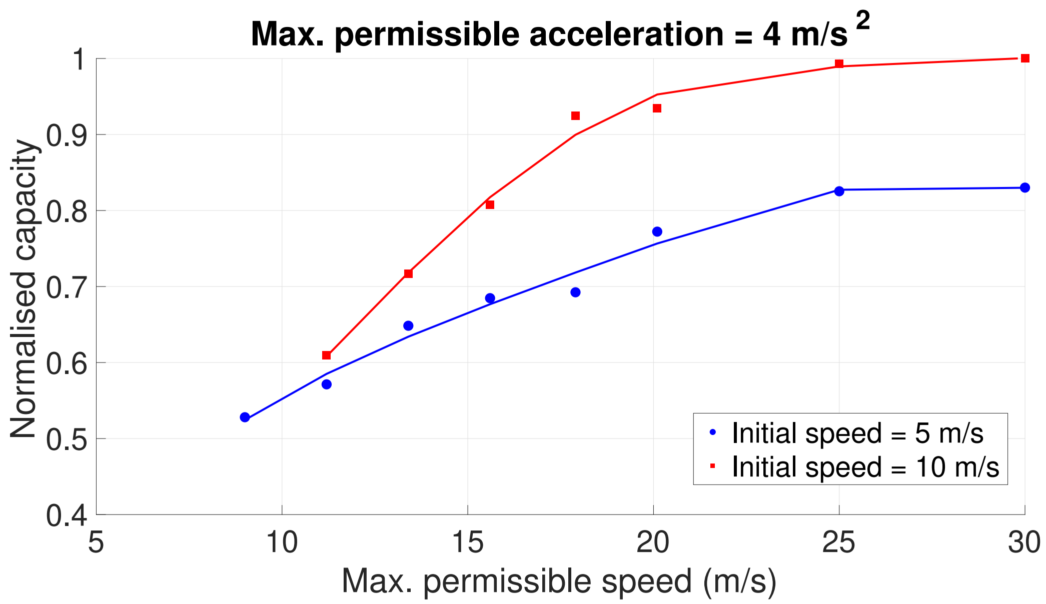
(a)



(b)

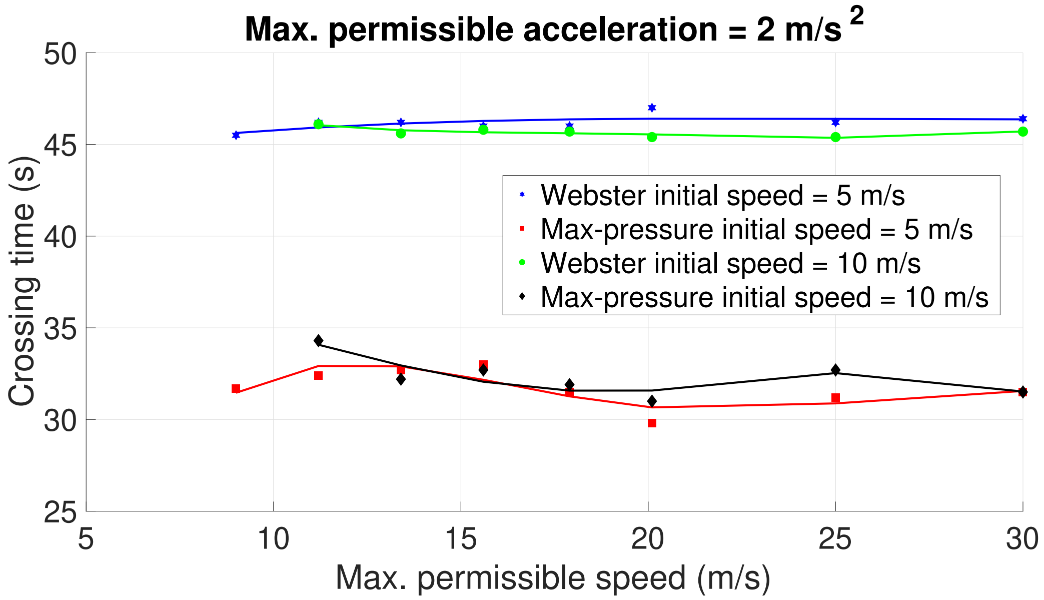


(c)

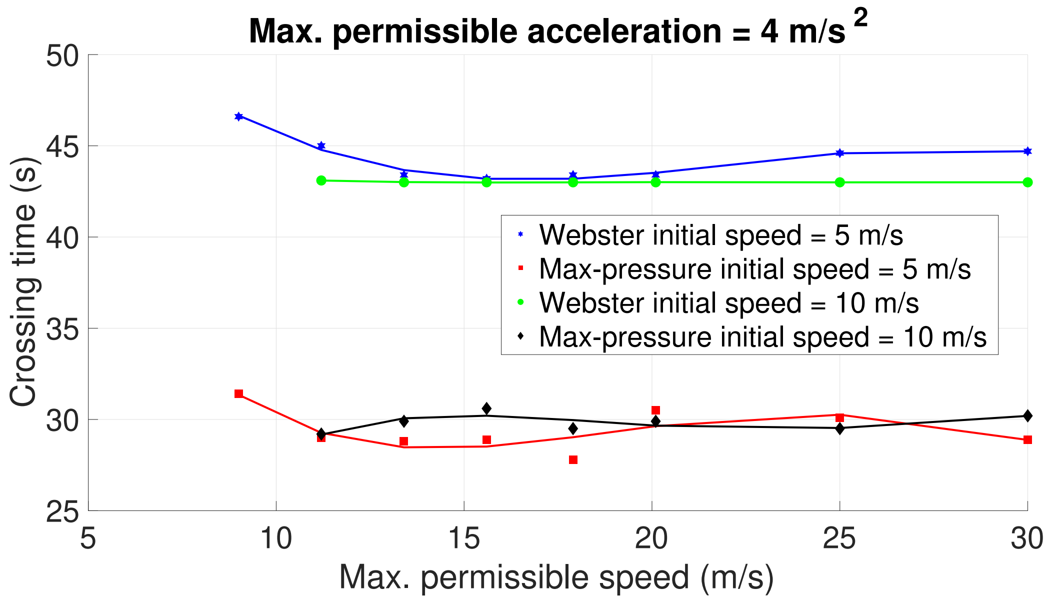


(d)

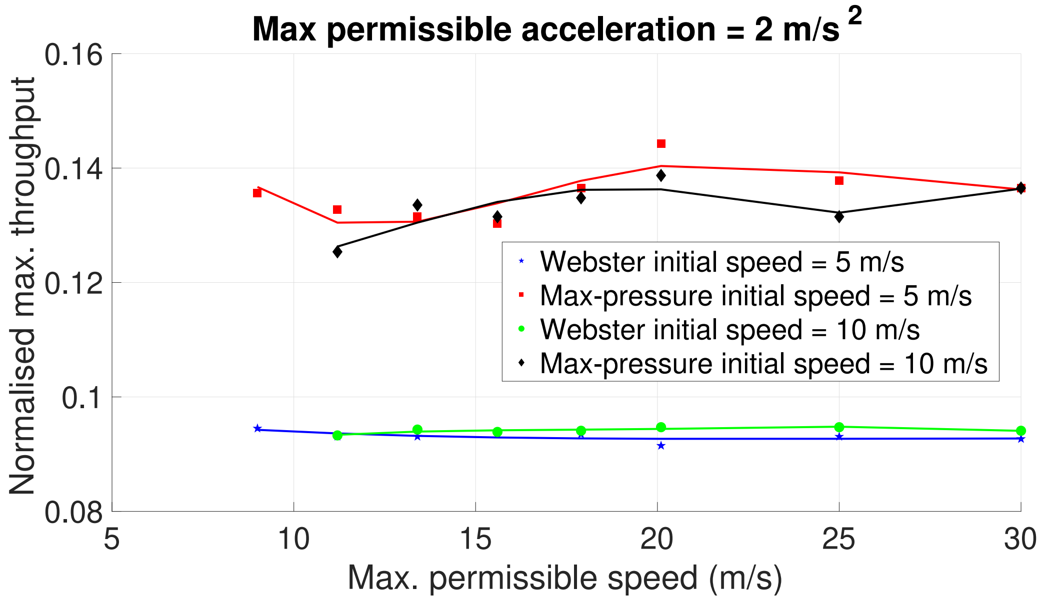
Figure 4.3: Variations of the crossing time when (a) Max. permissible acceleration is 2 m/s^2 and (b) Max. permissible acceleration is 4 m/s^2 and variations of capacity when (c) Max. permissible acceleration is 2 m/s^2 and (d) Max. permissible acceleration is 4 m/s^2 of the studied lane-free intersection over different values of the initial speed and the maximum permissible speed of CAVs. The solid lines are the corresponding fitted polynomials of order four, which show the variation trends.


Max. permissible acceleration = 2 m/s 2
Crossing time (s)
Max. permissible speed (m/s)
Webster initial speed = 5 m/s
Max-pressure initial speed = 5 m/s
Webster initial speed = 10 m/s
Max-pressure initial speed = 10 m/s
50
45
40
35
30
25
5
10
15
20
25
30

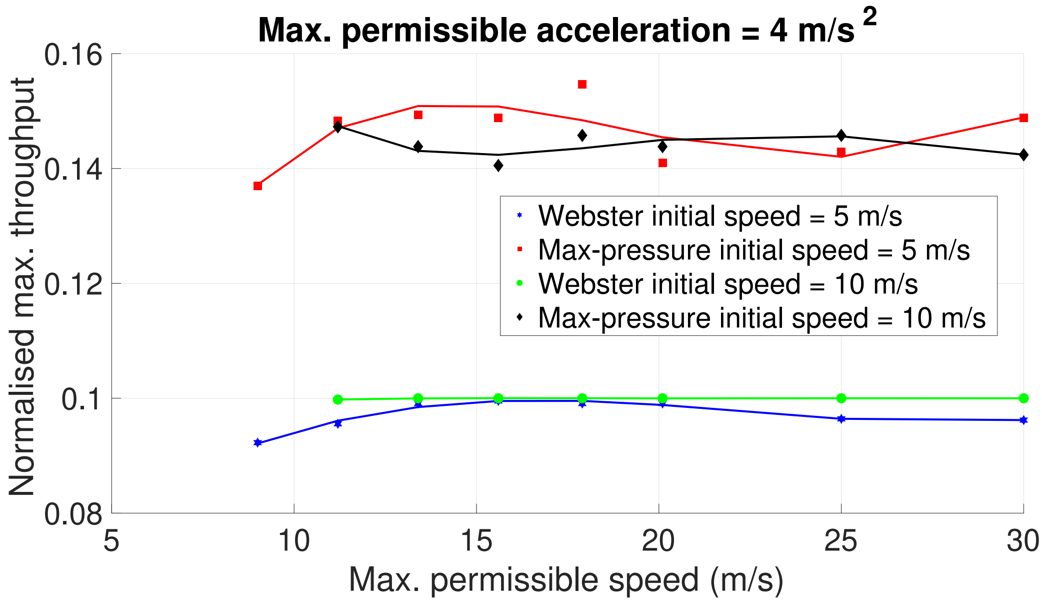
(a)


Max. permissible acceleration = 4 m/s 2
Crossing time (s)
Max. permissible speed (m/s)
Webster initial speed = 5 m/s
Max-pressure initial speed = 5 m/s
Webster initial speed = 10 m/s
Max-pressure initial speed = 10 m/s
50
45
40
35
30
25
5
10
15
20
25
30

(b)



(c)



(d)

Figure 4.4: Variations of the crossing time when (a) Max. permissible acceleration is 2 m/s^2 and (b) Max. permissible acceleration is 4 m/s^2 and variations of maximum throughput when (c) Max. permissible acceleration is 2 m/s^2 and (d) Max. permissible acceleration is 4 m/s^2 of the signalised intersection for different values of the initial speed and the maximum permissible speed of the crossing HVs. The solid lines show trends of the variation as polynomials of order four.

4.4 Summary

This chapter introduced a measure to represent the capacity of a given intersection when CAVs are crossing in a lane-free order, along with an algorithm to calculate the measure. The measure gradually calculates the capacity using the developed minimum-time lane-free crossing algorithm in Chapter 3. The presented results show that the lane-free crossing of CAVs improves capacity of an intersection by, respectively, 127% and 36% as compared to capacity of the signalised crossing of human drivers and CAVs through the same intersection which are calculated using highway capacity manual. In addition, the results of this work also provides a benchmark to evaluate the performance of the algorithms to collaboratively cross CAVs through intersections. A sensitivity analysis is also presented showing that in contrast to the maximum throughput of signalised intersections the capacity of lane-free intersections varies with changes in maximum permissible acceleration and speed and the initial speed of CAVs.

Chapter 5

Summary, Conclusions and Future Work

This chapter presents a detailed summary of the work in this thesis followed by the conclusions and direction for future works.

5.1 Summary

The work in this thesis is a research on design and simulation of connected and autonomous vehicles when passing through intersections in a lane-free order. Lane-free crossing takes the advantage of CAVs' collaboration to make use of all segments of roads and intersections which has shown in this study that it reduces energy consumption and increases traffic throughput. The research has successfully achieved all of its objectives in chapter 1.

The majority of previous studies employed a reservation-based strategy to control CAVs crossing intersections which restrain the motion of CAVs from leaving the lanes. Chapter 3 of this thesis, on the other hands, formulates and solves the lane-free crossing problem of CAVs through intersections as a minimum-time OCP since the final time is an unknown variable that needs to be minimised along with energy consumption while avoiding collisions. However, lane-free crossing problem is challenging as it involves non-differentiable collision avoidance constraints.

The non-differentiable constraints of CAVs avoiding collision with each other and with road boundaries are smoothened by applying dual problem theory of convex optimisation. The developed smooth OCP is then solved using CasADi and IPOPT and was tested within three different test scenarios.

Performance of the proposed algorithm is compared against that of the state-of-the-art reservation-based and lane-free methods. Ignoring the energy consumption, the results show that the crossing time of CAVs controlled by the proposed strategy is respectively, up to 65% and 54% less than the crossing time of CAVs controlled by the reservation-based and lane-free benchmark methods. However, such improvement was achieved at the cost of consuming more fuel. A similar comparison indicates that the proposed lane-free strategy reduces crossing time by up to 52% as compare to the reservation-based strategy while consuming the same amount of energy. Moreover, the results show that passengers are more comfortable with the lane-free crossing because of an acceleration profile which is far less than the maximum permissible value, and converges to zero at the destination.

Performance of the proposed lane-free algorithm is also evaluated for more complex scenarios with more number of CAVs. It is shown that crossing time of CAVs through an intersection is almost fixed to constant value regardless of the number of crossing CAVs.

Chapter 4 measures capacity improvements of intersections by the lane-free crossing of CAVs as compared to the signalised crossing. To the best knowledge of the author, there is no previous analysis that objectively quantifies such improvement. It is worth noting that: i) the conventional capacity measures are not applicable to the lane-free crossing; and ii) the crossing performance of CAVs depends on the collaborative behaviour of the vehicles and not the performance of either the traffic light controller (as in conventional intersections) or individual vehicle (as in autonomous vehicles without such collaborative behaviour). Chapter 4 presents a novel framework to evaluate capacity of intersections when CAVs are crossing in a lane-free order. The framework includes a measure and the minimum-time optimal lane-free algorithm developed in Chapter 3 to calculate the measure. It is shown that the capacity of a lane-free intersection with CAVs is, respectively, 127% and 36% higher than the capacity of a signalised intersection with CAVs and HVs. This massive improvement is because of shorter safety distance of CAVs and their ability to travel in a lane-free and signal-free order. Chapter 4 also presents a sensitivity analysis of capacity and crossing time of lane-free intersections with respect to the maximum permissible acceleration and speed, and the initial speed of CAVs.

5.2 Conclusions

The overall conclusions considering the objectives of the thesis are:

- The proposed lane-free crossing is a promising method for CAVs to cross intersections. This method improves the crossing time of a junction by an average of 40%

as compared to the state-of-the-art reservation-based method, whilst consuming the same amount of energy.

- The minimum crossing time of CAVs through intersections in a lane-free order is fixed to a constant value regardless of the number of CAVs up to reaching the capacity of the junction. In fact the crossing time of lane-free intersections is dependent to the layout of the junction and the CAV that is travelling the longest distance rather than the number of crossing vehicles as in signalised intersections.

- The proposed lane-free algorithm finds the minimum crossing time of CAVs which is very close to its theoretical limit. This makes the results of the proposed algorithm a suitable benchmark to evaluate the performance of other strategies that control CAVs to cross intersections.

- Lane-free crossing of CAVs improves capacity of an intersection by, respectively, 127% and 36% as compared to a signalised crossing of CAVs and HVs.

- A sensitivity analysis indicates that, unlike the signalised intersections, the maximum throughput of the lane-free crossing of CAVs is improved by increasing the initial speed, and maximum permissible speed and acceleration of the vehicles.

5.3 Future work

The following research directions are considered as future works:

- The computational complexity of solving the proposed OCP is of the order of $O(e^{0.13N})$, where N is the number of CAVs passing through the intersection. It is interesting to realise the proposed strategy as a decentralised algorithm over a receding horizon that can be applied to the real-time applications. In addition, such approach can take into account the uncertainties of measurements and models.

- Although the vehicle model employed for the OCP formulation is sufficient for lane-free movement, this model is yet non-linear. Future work will investigate vehicle models that take into account the lateral behaviour of CAVs for lane-free movement while , at least locally, not involve non-linearity.

- The gain parameters of the objective function of the OCP (e.g., crossing time α, acceleration γ and Lagrange term $\mathbf{Q}$) can be systematically tuned to improve the convergence and accuracy of the solution. Moreover, the gain of Lagrange term $\mathbf{Q}$, needs to be designed carefully as it is different for the elements (e.g., $x(t)$, $y(t)$ and $\theta(t)$) at each sampling time.

- The provided analysis of a single lane-free intersection can be extended to the case with multiple intersections and consider more factors such as passenger comfort into the measurement of capacity.

Appendix A

Main parts of the developed solver

The model parameters given in Table 3.1 are needed to initialise the states and control variables of the system. The following for loops create equations to initialise all the states and control signals of N_v vehicle:

```
% Declare model states and controls
for i = 1:N_v*noOfStates
    eval(sprintf('x%d = %s%s%s', i, 'SX.sym(''', sprintf('x%d', i), ''');'));
end

for i = 1:N_v*noOfControls
    eval(sprintf('u%d = %s%s%s', i, 'SX.sym(''', sprintf('u%d', i), ''');'));
end

for i = 1:N_v*noOfStates
    var = sprintf('x%d', i);
    x = [x; eval(var)];
end

for i = 1:N_v*noOfControls
    var = sprintf('u%d', i);
    u = [u; eval(var)];
end
```

To facilitate a user-friendly formulation of the problem the *Opti* [124] module of CasADi is employed:

```
opti = Opti(); %initialise Opti
```

Opti is designed to simplify the process of formulating optimization problems by providing a high-level functions such as *variable()*, *subject_to()* and *minimise()* for defining, respectively, variables, constraints and the objective.

T is the final time and is defined as a solo variable which is later is minimised:

```
T = opti.variable(); %setting final time as a variable
h = T/N_p; %sampling time
```

where h is the resulting sampling time.

To discretise the dynamics of the system, the collocation method explained in Section 2.6.3 is implemented. As discussed in Section 2.6.3, the collocation method uses the roots of a polynomial with orthogonality property in which Legendre polynomial of degree d is used here as follows:

```
tau = collocation_points(d, 'legendre');% Get collocation points
```

Using the roots, the coefficient of collocation equations denoted as C, coefficient of continuity equations represented as D and coefficient of the quadrature function presented by B, which are used later for discretisation, are obtained as follows:

```
[C,D,B] = collocation_coeff(tau);% Collocation linear maps
```

To discretise the dynamics and develop the resulting NLP, the following *for* loop is coded:

```
for k = 1:N_p-1
    Uk = opti.variable(N_v*noOfControls); %Control variables
    for i=1:N_v %boundary conditions for the control inputs
        opti.subject_to(...);
    end

    Xc = opti.variable(N_v*noOfStates, d); %state variables at collocations points
    for i=1:N_v %boundary conditions for states at collocation points
       opti.subject_to(...);
    end

    Xk = opti.variable(N_v*noOfStates); %state variables at end of interval
    for i=1:N_v %boundary conditions for states variables
       opti.subject_to(...);
    end

    lambda = opti.variable(N_v*length(b),N_v); %dual variable
    s = opti.variable(N_v*(N_v-1)/2,2); %dual variable

    for i=1:N_v %collision avoidance constrains of vehicles
       opti.subject_to(...);
    end
```

```
    lambda_rd = opti.variable((N_v+4)*4,N_v+4); %dual variable for roads
    s_rd = opti.variable(N_v*size(A,2),4); %dual variable for roads

    for i=1:N_v %collision avoidance constrains of vehicles and road boundaries
       opti.subject_to(...);
    end

    [ode, quad] = f(Xc, Uk); %evaluate ODE right-hand-side at collocations
    J = J + quad*B*h; %add contribution to quadrature function
    Z = [Xk Xc]; %get interpolating points of collocation polynomial
    Pidot = Z*C;%get slope of interpolating polynomial (normalized)
    opti.subject_to(Pidot == h*ode); %match with ODE right-hand-side
    Xk_end = Z*D; %state at end of collocation interval
    opti.subject_to(Xk_end==Xk) %continuity constraints
end
```

In the above codes, the boundary conditions of control inputs and state variables are initially defined and stored in the Opti object. It must be noted that the boundary conditions are enforced for both the collocation points Xc and for the state intervals Xk. The collision avoidance constrains, on the other hand, are only enforced for the state variables Xk and not for the collocation point variables Xc. This is because enforcing the collision avoidance constrains for both Xk and Xc significantly increases the computational time while enforcing them just for the state variables Xk is sufficient to obtain collision-free trajectories.

The stage cost of (3.9) and the dynamics given in (2.1) are symbolically formulated as respectively L and $xdot$ using the aforementioned defined states and control variables. L and xdot are both set as the outputs of CasADi defined function as follows:

```
for i = 1:N_v
    xdot = [xdot; eval(sprintf('x%d*cos(x%d)', i*noOfStates-(noOfStates-4),
           i*noOfStates-(noOfStates-3)));eval(sprintf('x%d*sin(x%d)',
           i*noOfStates-(noOfStates-4), i*noOfStates-(noOfStates-3)));
           eval(sprintf('x%d', i*noOfStates-(noOfStates-5)));
           eval(sprintf('u%d', i*noOfControls-(noOfControls-1)));
           eval(sprintf('(%f/(%f*x%d))*x%d+(%f/%f)*x%d+(%f/%f)*u%d',Nr,Iz,
           i*noOfStates-(noOfStates-4),i*noOfStates-(noOfStates-5),Nb,Iz,
           i*noOfStates,Nphi,Iz,i*noOfControls));
           eval(sprintf('(%f/(%f*x%d*x%d)-%f)*x%d+%f/(%f*x%d)*
           x%d+%f/(%f*x%d)*u%d',Yr,m,i*noOfStates-(noOfStates-4),
           i*noOfStates-(noOfStates-4),1,i*noOfStates-(noOfStates-5),Yb,m,
```

```
            i*noOfStates-(noOfStates-4),i*noOfStates,Yphi,m,
            i*noOfStates-(noOfStates-4),i*noOfControls))];
end

for i = 1:N_v
    L =  sprintf('%s + %f*(x%d-%f)^2+%f*(x%d-%f)^2+%f*(x%d-%f)^2+0*(u%d)^2',
    p,G(1),i*noOfStates-(noOfStates-1),Xk_dest(i*noOfStates-(noOfStates-1)),
    G(2), i*noOfStates-(noOfStates-2), Xk_dest(i*noOfStates-(noOfStates-2)),
    G(3), i*noOfStates-(noOfStates-3), Xk_dest(i*noOfStates-(noOfStates-3)),
    i*noOfControls-(noOfControls-1));
end
```

where G represent the gain matrix and L and $xdot$ are combined within a symbolic function f as follows:

```
f = Function('f', {x, u}, {xdot, L});
```

The calculated stage cost L is summed up to construct the overall cost J which is later minimised.

Using the codes above, the OCP (3.8) is converted to a large-scale NLP which is then solve by the IPOPT [111] as the state-of-the-art NLP solver:

```
s_opts = struct("linear_solver",'ma27');
opti.solver('ipopt',s_opts);
```

Appendix B

List of publications

1- Amouzadi, M., Orisatoki, M. O., Dizqah, A. M. (2022) ‘Optimal Lane-Free Crossing of CAVs through Intersections’. IEEE Transactions on Vehicular Technology. doi: 10.1109/TVT.2022.3207054.

2- Amouzadi, M., Orisatoki, M. O., Dizqah, A. M. (2022) ‘Capacity Analysis of Intersections When CAVs Are Crossing in a Collaborative and Lane-Free Order’, Future Transportation, 2(3), 698–710. doi: 10.3390/futuretransp2030039.

3- Amouzadi, M., Orisatoki, M. O., Dizqah, A. M. (2022) ‘Lane-Free Crossing of CAVs through Intersections as a Minimum-Time Optimal Control Problem’, IFAC-PapersOnLine, 55(14), 28–33. doi: 10.1016/j.ifacol.2022.07.578.

Contributions to Other Projects

M., Orisatoki, Amouzadi, M, Dizqah, A. M. (2022) ’A Heuristic Informative Path Planning Algorithm for Mapping the Unknown Non-Convex Areas’ IEEE Robotics and Automation Letters. - Ready to submit.

Bibliography

[1] A. A. Malikopoulos, L. Beaver, and I. V. Chremos, "Optimal time trajectory and coordination for connected and automated vehicles," *Automatica*, vol. 125, p. 109469, 2021.

[2] B. Li, Y. Zhang, N. Jia, and X. Peng, "Autonomous intersection management over continuous space: A microscopic and precise solution via computational optimal control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 17071–17076, 2020.

[3] Y. Cao, T. Griffon, F. Fahrenkrog, M. Schneider, F. Naujoks, F. Tango, S. Wolter, A. Knapp, Y. Page, J. Mallada, G. Dominioni, E. Demirtzis, M. Giorelli, S. Fabello, F. Frey, Q. Feng, O. Brunnegård, A. Kucewicz, S. Whitehouse, and U. Eberle, "L3pilot - code of practice for the development of automated driving functions," 04 2021.

[4] A. Moradinegade Dizqah, "Non-linear model predictive energy management," 2014.

[5] L. Biegler, *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes.* MOS-SIAM Series on Optimization, Society for Industrial and Applied Mathematics, 2010.

[6] S. He, F. Ding, C. Lu, and Y. Qi, "Impact of connected and autonomous vehicle dedicated lane on the freeway traffic efficiency," *European Transport Research Review*, vol. 14, no. 1, pp. 1–14, 2022.

[7] M. Amouzadi, M. O. Orisatoki, and A. M. Dizqah, "Lane-free crossing of cavs through intersections as a minimum-time optimal control problem," *IFAC-PapersOnLine*, vol. 55, no. 14, pp. 28–33, 2022. 11th IFAC Symposium on Intelligent Autonomous Vehicles IAV 2022.

[8] M. Amouzadi, M. O. Orisatoki, and A. M. Dizqah, "Optimal lane-free crossing of cavs through intersections," *IEEE Transactions on Vehicular Technology*, 2022.

[9] M. Amouzadi, M. O. Orisatoki, and A. M. Dizqah, "Capacity analysis of intersections when cavs are crossing in a collaborative and lane-free order," *Future Transportation*, vol. 2, no. 3, pp. 698–710, 2022.

[10] P. Kopelias, E. Demiridi, K. Vogiatzis, A. Skabardonis, and V. Zafiropoulou, "Connected & autonomous vehicles–environmental impacts–a review," *Science of The Total Environment*, vol. 712, p. 135237, 2020.

[11] M. Taiebat, A. L. Brown, H. R. Safford, S. Qu, and M. Xu, "A review on energy, environmental, and sustainability implications of connected and automated vehicles," *Environmental science & technology*, vol. 52, no. 20, pp. 11449–11465, 2018.

[12] A. Brown, B. Repac, and J. Gonder, "Autonomous vehicles have a wide range of possible energy impacts," tech. rep., NREL, University of Maryland, 2013.

[13] J. Gonder, M. Earleywine, and W. Sparks, "Analyzing vehicle fuel saving opportunities through intelligent driver feedback," *SAE International Journal of Passenger Cars-Electronic and Electrical Systems*, vol. 5, no. 2012-01-0494, pp. 450–461, 2012.

[14] D. Schrank, B. Eisele, T. Lomax, and J. Bak, "2015 urban mobility scorecard," 2015.

[15] A. Foxx and P. John, "State transportation statistics," in *U.S. Department of Transportation*, 2012.

[16] M. Rahman, M. R. Islam, M. Chowdhury, and T. Khan, "Development of a connected and automated vehicle longitudinal control model," *arXiv preprint arXiv:2001.00135*, 2020.

[17] P. Tallapragada and J. Cortés, "Hierarchical-distributed optimized coordination of intersection traffic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 2100–2113, 2019.

[18] A. A. Malikopoulos, C. G. Cassandras, and Y. J. Zhang, "A decentralized energy-optimal control framework for connected automated vehicles at signal-free intersections," *Automatica*, vol. 93, pp. 244–256, 2018.

[19] S. D. Kumaravel, A. A. Malikopoulos, and R. Ayyagari, "Decentralized cooperative merging of platoons of connected and automated vehicles at highway on-ramps," in *2021 American Control Conference (ACC)*, pp. 2055–2060, IEEE, 2021.

[20] T. Shim, G. Adireddy, and H. Yuan, "Autonomous vehicle collision avoidance system using path planning and model-predictive-control-based active front steering and wheel torque control," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of automobile engineering*, vol. 226, no. 6, pp. 767–778, 2012.

[21] I. Batkovic, M. Zanon, M. Ali, and P. Falcone, "Real-time constrained trajectory planning and vehicle control for proactive autonomous driving with road users," in *2019 18th European Control Conference (ECC)*, pp. 256–262, IEEE, 2019.

[22] Y. Rasekhipour, A. Khajepour, S.-K. Chen, and B. Litkouhi, "A potential field-based model predictive path-planning controller for autonomous road vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1255–1267, 2016.

[23] G. Schildbach and F. Borrelli, "Scenario model predictive control for lane change assistance on highways," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, pp. 611–616, IEEE, 2015.

[24] W. Milliken and D. Milliken, *Race Car Vehicle Dynamics.* Premiere Series, SAE International, 1995.

[25] P. Typaldos, M. Papageorgiou, and I. Papamichail, "Optimization-based path-planning for connected and non-connected automated vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 134, p. 103487, 2022.

[26] S. H. Kim and R. Bhattacharya, "Motion planning in obstacle rich environments," *Journal of Aerospace Computing, Information, and Communication*, vol. 6, no. 7, pp. 433–450, 2009.

[27] A. Rusu, S. Moreno, Y. Watanabe, M. Rognant, and M. Devy, "State lattice generation and nonholonomic path planning for a planetary exploration rover," in *65th International Astronautical Congress 2014 (IAC 2014)*, vol. 2, p. 953, 2014.

[28] M. Likhachev and D. Ferguson, "Planning long dynamically feasible maneuvers for autonomous vehicles," *The International Journal of Robotics Research*, vol. 28, no. 8, pp. 933–945, 2009.

[29] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research direc-

tions," *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 416–442, 2015.

[30] R. Kala and K. Warwick, "Multi-level planning for semi-autonomous vehicles in traffic scenarios based on separation maximization," *Journal of Intelligent & Robotic Systems*, vol. 72, no. 3-4, pp. 559–590, 2013.

[31] J. Bohren, T. Foote, J. Keller, A. Kushleyev, D. Lee, A. Stewart, P. Vernaza, J. Derenick, J. Spletzer, and B. Satterfield, "Little ben: The ben franklin racing team's entry in the 2007 darpa urban challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 598–614, 2008.

[32] S. Panich, "The shortest path with intelligent algorithm," *Journal of Mathematics and Statistics*, vol. 6, no. 3, pp. 276–278, 2010.

[33] S. Kammel, J. Ziegler, B. Pitzer, M. Werling, T. Gindele, D. Jagzent, J. Schröder, M. Thuy, M. Goebl, F. v. Hundelshausen, *et al.*, "Team annieway's autonomous system for the 2007 darpa urban challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 615–639, 2008.

[34] A. Stentz, "Optimal and efficient path planning for partially known environments," in *Intelligent unmanned ground vehicles*, pp. 203–220, Springer, 1997.

[35] A. Nash, K. Daniel, S. Koenig, and A. Felner, "Theta^*: Any-angle path planning on grids," in *AAAI*, vol. 7, pp. 1177–1183, 2007.

[36] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, *et al.*, "Junior: The stanford entry in the urban challenge," *Journal of field Robotics*, vol. 25, no. 9, pp. 569–597, 2008.

[37] M. Pivtoraiko and A. Kelly, "Efficient constrained path planning via search in state lattices," in *International Symposium on Artificial Intelligence, Robotics, and Automation in Space*, pp. 1–7, Munich Germany, 2005.

[38] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on intelligent transportation systems*, vol. 17, no. 4, pp. 1135–1145, 2015.

[39] Y. Gao, A. Gray, J. V. Frasch, T. Lin, E. Tseng, J. K. Hedrick, and F. Borrelli, "Spatial predictive control for agile semi-autonomous ground vehicles," in *Proceedings of the 11th international symposium on advanced vehicle control*, no. 2, pp. 1–6, 2012.

[40] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[41] S. J. Anderson, S. C. Peters, T. E. Pilutti, and K. Iagnemma, "An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios," *International Journal of Vehicle Autonomous Systems*, vol. 8, no. 2-4, pp. 190–216, 2010.

[42] Z. Htike, G. Papaioannou, E. Siampis, E. Velenis, and S. Longo, "Minimisation of motion sickness in autonomous vehicles," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1135–1140, IEEE, 2020.

[43] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, and D. Hrovat, "A hierarchical model predictive control framework for autonomous ground vehicles," in *2008 American Control Conference*, pp. 3719–3724, IEEE, 2008.

[44] P. Liu and Ü. Özgüner, "Predictive control of a vehicle convoy considering lane change behavior of the preceding vehicle," in *2015 American Control Conference (ACC)*, pp. 4374–4379, IEEE, 2015.

[45] P. Falcone, F. Borrelli, J. Asgari, H. Tseng, and D. Hrovat, "Low complexity mpc schemes for integrated vehicle dynamics control problems," in *9th international symposium on advanced vehicle control*, 2008.

[46] M. Werling and D. Liccardo, "Automatic collision avoidance using model-predictive online optimization," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pp. 6309–6314, IEEE, 2012.

[47] Y. Huang, H. Wang, A. Khajepour, H. Ding, K. Yuan, and Y. Qin, "A novel local motion planning framework for autonomous vehicles based on resistance network and model predictive control," *IEEE Transactions on Vehicular Technology*, 2019.

[48] C. Liu, S. Lee, S. Varnhagen, and H. E. Tseng, "Path planning for autonomous vehicles using model predictive control," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 174–179, IEEE, 2017.

[49] M. Zhou, Y. Yu, and X. Qu, "Development of an efficient driving strategy for connected and automated vehicles at signalized intersections: A reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[50] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1343–1350, IEEE, 2017.

[51] F. Ye, S. Zhang, P. Wang, and C.-Y. Chan, "A survey of deep reinforcement learning algorithms for motion planning and control of autonomous vehicles," in *2021 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1073–1080, IEEE, 2021.

[52] S. J. Russell, *Artificial intelligence a modern approach.* Pearson Education, Inc., 2010.

[53] R. S. Sutton, A. G. Barto, *et al.*, "Reinforcement learning," *Journal of Cognitive Neuroscience*, vol. 11, no. 1, pp. 126–134, 1999.

[54] J. V. Frasch, A. Gray, M. Zanon, H. J. Ferreau, S. Sager, F. Borrelli, and M. Diehl, "An auto-generated nonlinear mpc algorithm for real-time obstacle avoidance of ground vehicles," in *2013 European Control Conference (ECC)*, pp. 4136–4141, IEEE, 2013.

[55] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 285–292, IEEE, 2017.

[56] J. Rios-Torres and A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1066–1077, 2016.

[57] Y. Zhang, A. A. Malikopoulos, and C. G. Cassandras, "Decentralized optimal control for connected automated vehicles at intersections including left and right turns," in

2017 IEEE 56th Annual Conference on Decision and Control (CDC), pp. 4428–4433, IEEE, 2017.

[58] B. Xu, S. E. Li, Y. Bian, S. Li, X. J. Ban, J. Wang, and K. Li, "Distributed conflict-free cooperation for multiple connected vehicles at unsignalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 93, pp. 322–334, 2018.

[59] M. Di Vaio, P. Falcone, R. Hult, A. Petrillo, A. Salvi, and S. Santini, "Design and experimental validation of a distributed interaction protocol for connected autonomous vehicles at a road intersection," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 9451–9465, 2019.

[60] L. Makarem and D. Gillet, "Fluent coordination of autonomous vehicles at intersections," in *2012 IEEE international conference on systems, man, and cybernetics (SMC)*, pp. 2557–2562, IEEE, 2012.

[61] A. Mirheli, L. Hajibabai, and A. Hajbabaie, "Development of a signal-head-free intersection control logic in a fully connected and autonomous vehicle environment," *Transportation Research Part C: Emerging Technologies*, vol. 92, pp. 412–425, 2018.

[62] A. Mirheli, M. Tajalli, L. Hajibabai, and A. Hajbabaie, "A consensus-based distributed trajectory control in a signal-free intersection," *Transportation research part C: emerging technologies*, vol. 100, pp. 161–176, 2019.

[63] H. Xu, C. G. Cassandras, L. Li, and Y. Zhang, "Comparison of cooperative driving strategies for cavs at signal-free intersections," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[64] M. Cai, Q. Xu, C. Chen, J. Wang, K. Li, J. Wang, and X. Wu, "Multi-lane unsignalized intersection cooperation with flexible lane direction based on multi-vehicle formation control," *IEEE Transactions on Vehicular Technology*, 2022.

[65] Z. He, L. Zheng, L. Lu, and W. Guan, "Erasing lane changes from roads: A design of future road intersections," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 2, pp. 173–184, 2018.

[66] B. Li, Y. Zhang, Y. Zhang, N. Jia, and Y. Ge, "Near-optimal online motion planning of connected and automated vehicles at a signal-free and lane-free intersection," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1432–1437, IEEE, 2018.

[67] A. Colombo and D. Del Vecchio, "Efficient algorithms for collision avoidance at intersections," in *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, pp. 145–154, 2012.

[68] C. Liu, C.-W. Lin, S. Shiraishi, and M. Tomizuka, "Distributed conflict resolution for connected autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 1, pp. 18–29, 2017.

[69] R. Hult, M. Zanon, S. Gros, and P. Falcone, "Optimal coordination of automated vehicles at intersections with turns," in *2019 18th European Control Conference (ECC)*, pp. 225–230, IEEE, 2019.

[70] R. Hult, M. Zanon, S. Gros, H. Wymeersch, and P. Falcone, "Optimisation-based coordination of connected, automated vehicles at intersections," *Vehicle System Dynamics*, vol. 58, no. 5, pp. 726–747, 2020.

[71] A. Katriniok, P. Kleibaum, and M. Joševski, "Distributed model predictive control for intersection automation using a parallelized optimization approach," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5940–5946, 2017.

[72] M. Kloock, P. Scheffe, S. Marquardt, J. Maczijewski, B. Alrifaee, and S. Kowalewski, "Distributed model predictive intersection control of multiple vehicles," in *2019 IEEE intelligent transportation systems conference (ITSC)*, pp. 1735–1740, IEEE, 2019.

[73] Y. Bian, S. E. Li, W. Ren, J. Wang, K. Li, and H. X. Liu, "Cooperation of multiple connected vehicles at unsignalized intersections: Distributed observation, optimization, and control," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 12, pp. 10744–10754, 2019.

[74] D. Elliott, W. Keen, and L. Miao, "Recent advances in connected and automated vehicles," *journal of traffic and transportation engineering (English edition)*, vol. 6, no. 2, pp. 109–131, 2019.

[75] X. Pan, B. Chen, S. A. Evangelou, and S. Timotheou, "Optimal motion control for connected and automated electric vehicles at signal-free intersections," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 2831–2836, IEEE, 2020.

[76] R. Molina-Masegosa and J. Gozalvez, "Lte-v for sidelink 5g v2x vehicular communications: A new 5g technology for short-range vehicle-to-everything communications," *IEEE Vehicular Technology Magazine*, vol. 12, no. 4, pp. 30–39, 2017.

[77] Y. Wu and F. Zhu, "Junction management for connected and automated vehicles: Intersection or roundabout?," *Sustainability*, vol. 13, no. 16, p. 9482, 2021.

[78] N. R. C. U. T. R. Board, *Highway Capacity Manual*. Transportation Research Board, National Research Council, 2000.

[79] S. Farivar, *Modeling capacity and delay at signalized intersections with channelized right-turn lanes considering the impact of blockage*. University of Nevada, Reno, 2015.

[80] N. Wu and S. Giuliani, "Capacity and delay estimation at signalized intersections under unsaturated flow condition based on cycle overflow probability," *Transportation research procedia*, vol. 15, pp. 63–74, 2016.

[81] M. Mohan and S. Chandra, "Capacity estimation of unsignalized intersections under heterogeneous traffic conditions," *Canadian Journal of Civil Engineering*, vol. 47, no. 6, pp. 651–662, 2020.

[82] R. Mohajerpoor and M. Ramezani, "Mixed flow of autonomous and human-driven vehicles: Analytical headway modeling and optimal lane management," *Transportation research part C: emerging technologies*, vol. 109, pp. 194–210, 2019.

[83] L. Zhao, A. A. Malikopoulos, and J. Rios-Torres, "On the traffic impacts of optimally controlled connected and automated vehicles," in *2019 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 882–887, IEEE, 2019.

[84] J. VanderWerf, S. Shladover, N. Kourjanskaia, M. Miller, and H. Krishnan, "Modeling effects of driver control assistance systems on traffic," *Transportation Research Record*, vol. 1748, no. 1, pp. 167–174, 2001.

[85] J. Ivanchev, A. Knoll, D. Zehe, S. Nair, and D. Eckhoff, "Potentials and implications of dedicated highway lanes for autonomous vehicles," *arXiv preprint arXiv:1709.07658*, 2017.

[86] İ. Altay, B. Aksun Güvenç, and L. Güvenç, "Lidar data analysis for time to headway determination in the drivesafe project field tests," *International journal of vehicular technology*, vol. 2013, 2013.

[87] C. Yu, W. Sun, H. X. Liu, and X. Yang, "Managing connected and automated vehicles at isolated intersections: From reservation-to optimization-based methods," *Transportation research part B: methodological*, vol. 122, pp. 416–435, 2019.

[88] L. Li, J. Gan, F. Yang, B. Ran, and J. Zhang, "A method to calculate the capacity of unsignalized intersection for connected and automated vehicles based on gap acceptance theory," in *CICTP 2019*, pp. 5600–5612, 2019.

[89] M. Papageorgiou, K.-S. Mountakis, I. Karafyllis, I. Papamichail, and Y. Wang, "Lane-free artificial-fluid concept for vehicular traffic," *Proceedings of the IEEE*, vol. 109, no. 2, pp. 114–121, 2021.

[90] K. Chavoshi and A. Kouvelas, "Distributed control for laneless and directionless movement of connected and automated vehicles," in *21st Swiss Transport Research Conference (STRC 2021)*, STRC, 2021.

[91] W. H. Kwon and S. Han, "Optimal controls on finite and infinite horizons: A review," *Receding Horizon Control: Model Predictive Control for State Models*, pp. 17–82, 2005.

[92] L. Grüne, "Nmpc without terminal constraints," *IFAC Proceedings Volumes*, vol. 45, no. 17, pp. 1–13, 2012.

[93] T. Lipp and S. Boyd, "Variations and extension of the convex–concave procedure," *Optimization and Engineering*, vol. 17, no. 2, pp. 263–287, 2016.

[94] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Functions of One Real Variable*, pp. 1–46. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993.

[95] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2004.

[96] M. Slater, "Lagrange multipliers revisited," in *Traces and emergence of nonlinear programming*, pp. 293–306, Springer, 2014.

[97] J. Thim, *Continuous nowhere differentiable functions.* 2003.

[98] G. Lars and P. Jürgen, "Nonlinear model predictive control theory and algorithms," 2011.

[99] R. Findeisen and F. Allgöwer, "An introduction to nonlinear model predictive control," in *21st Benelux meeting on systems and control*, vol. 11, pp. 119–141, Citeseer, 2002.

[100] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[101] B. Lincoln and A. Rantzer, "Relaxing dynamic programming," *IEEE Transactions on Automatic Control*, vol. 51, no. 8, pp. 1249–1260, 2006.

[102] N. Kim, S. Cha, and H. Peng, "Optimal control of hybrid electric vehicles based on pontryagin's minimum principle," *IEEE Transactions on control systems technology*, vol. 19, no. 5, pp. 1279–1287, 2010.

[103] M. Diehl, H. G. Bock, H. Diedam, and P.-B. Wieber, "Fast direct multiple shooting algorithms for optimal robot control," in *Fast motions in biomechanics and robotics*, pp. 65–93, Springer, 2006.

[104] L. T. Biegler, "An overview of simultaneous strategies for dynamic optimization," *Chemical Engineering and Processing: Process Intensification*, vol. 46, no. 11, pp. 1043–1053, 2007.

[105] C. Kirches, L. Wirsching, H. Bock, and J. Schlöder, "Efficient direct multiple shooting for nonlinear model predictive control on long horizons," *Journal of Process Control*, vol. 22, no. 3, pp. 540–550, 2012.

[106] H. G. Bock and K.-J. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," *IFAC Proceedings Volumes*, vol. 17, no. 2, pp. 1603–1608, 1984.

[107] S. Munzir, V. Halfiani, *et al.*, "An optimal control solution using multiple shooting method," *Bulletin of Mathematics*, vol. 4, no. 02, pp. 143–160, 2012.

[108] L. T. Biegler, "Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation," *Computers & chemical engineering*, vol. 8, no. 3-4, pp. 243–247, 1984.

[109] A. Huber, M. Gerdts, and E. Bertolazzi, "Structure exploitation in an interior-point method for fully discretized, state constrained optimal control problems," *Vietnam Journal of Mathematics*, vol. 46, no. 4, pp. 1089–1113, 2018.

[110] S. Kameswaran and L. T. Biegler, "Simultaneous dynamic optimization strategies: Recent advances and challenges," *Computers & Chemical Engineering*, vol. 30, no. 10-12, pp. 1560–1575, 2006.

[111] L. T. Biegler and V. M. Zavala, "Large-scale nonlinear programming using ipopt: An integrating framework for enterprise-wide dynamic optimization," *Computers & Chemical Engineering*, vol. 33, no. 3, pp. 575–582, 2009.

[112] R. Bartlett, A. Wachter, and L. Biegler, "Active set vs. interior point strategies for model predictive control," in *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334)*, vol. 6, pp. 4229–4233, IEEE, 2000.

[113] I. S. Duff, "Ma57—a code for the solution of sparse symmetric definite and indefinite systems," *ACM Transactions on Mathematical Software (TOMS)*, vol. 30, no. 2, pp. 118–144, 2004.

[114] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster, "Mumps: a general purpose distributed memory sparse solver," in *International Workshop on Applied Parallel Computing*, pp. 121–130, Springer, 2000.

[115] R. H. Byrd, J. C. Gilbert, and J. Nocedal, "A trust region method based on interior point techniques for nonlinear programming," *Mathematical programming*, vol. 89, no. 1, pp. 149–185, 2000.

[116] J. Andersson, J. Åkesson, and M. Diehl, "Dynamic optimization with casadi," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pp. 681–686, IEEE, 2012.

[117] R. D. Neidinger, "Introduction to automatic differentiation and matlab object-oriented programming," *SIAM review*, vol. 52, no. 3, pp. 545–563, 2010.

[118] L. B. Rall and G. F. Corliss, "An introduction to automatic differentiation," *Computational Differentiation: Techniques, Applications, and Tools*, vol. 89, 1996.

[119] J. Andersson, J. Åkesson, and M. Diehl, "Casadi: A symbolic package for automatic differentiation and optimal control," in *Recent advances in algorithmic differentiation*, pp. 297–307, Springer, 2012.

[120] B. Houska, H. J. Ferreau, and M. Diehl, "An auto-generated real-time iteration algorithm for nonlinear mpc in the microsecond range," *Automatica*, vol. 47, no. 10, pp. 2279–2285, 2011.

[121] H. J. Ferreau, S. Almér, R. Verschueren, M. Diehl, D. Frick, A. Domahidi, J. Jerez, G. Stathopoulos, and C. Jones, "Embedded optimization methods for industrial automatic control," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 13194–13209, 2017.

[122] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 972–983, 2021.

[123] R. Firoozi, L. Ferranti, X. Zhang, S. Nejadnik, and F. Borrelli, "A distributed multi-robot coordination algorithm for navigation in tight environments," *arXiv preprint arXiv:2006.11492*, 2020.

[124] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[125] M. S. Chaudhry and P. Ranjitkar, *Capacity analysis of signalised intersection using micro-simulation*. PhD thesis, University of Auckland, 2009.

[126] P. Chauhan Boski, J. Joshi Gaurang, and P. Purnima, "Microscopic traffic simulation approach to evaluate driving parameters at influence zone of the intersection," in *Intelligent Infrastructure in Transportation and Management*, pp. 13–26, Springer, 2022.

[127] A. Ghiasi, O. Hussain, Z. S. Qian, and X. Li, "A mixed traffic capacity analysis and lane management model for connected automated vehicles: A markov chain method," *Transportation Research Part B: Methodological*, vol. 106, pp. 266–292, 2017.

[128] M. Martin-Gasulla, P. Sukennik, and J. Lohmiller, "Investigation of the impact on throughput of connected autonomous vehicles with headway based on the leading vehicle type," *Transportation Research Record*, vol. 2673, no. 5, pp. 617–626, 2019.

[129] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 177–195, 2013.

[130] F. V. Webster, "Traffic signal settings," tech. rep., 1958.

[131] W. Genders and S. Razavi, "An open-source framework for adaptive traffic signal control," *arXiv preprint arXiv:1909.00395*, 2019.

[132] C. Gershenson, "Self-organizing traffic lights," *arXiv preprint nlin/0411066*, 2004.

[133] S.-B. Cools, C. Gershenson, and B. D'Hooghe, "Self-organizing traffic lights: A realistic simulation," in *Advances in applied self-organizing systems*, pp. 45–55, Springer, 2013.

[134] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[135] N. Casas, "Deep deterministic policy gradient for urban traffic light control," *arXiv preprint arXiv:1703.09035*, 2017.