

TECHNICAL UNIVERSITY OF CRETE



DOCTORAL THESIS

Effect of Automated Vehicles on Highway Traffic Flow and Signalized Junctions

Author:
Panagiotis TYPALDOS

Supervisor:
Prof. Markos PAPAGEORGIU

*A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy*

in the

Dynamic Systems & Simulations Laboratory
School of Production, Engineering & Management

December 2022

The present thesis is approved by the following jury:

Advisory Committee:

Markos Papageorgiou (Supervisor)

Professor, School of Production Engineering and Management
Technical University of Crete, Chania, Greece

Ioannis Papamichail (Member of committee in charge)

Professor, School of Production Engineering and Management
Technical University of Crete, Chania, Greece

Ioannis Nikolos (Member of committee in charge)

Professor, School of Production Engineering and Management
Technical University of Crete, Chania, Greece

Thesis Committee:

Elias Kosmatopoulos

Professor, Department of Electrical and Computer Engineering
Democritus University of Thrace, Xanthi, Greece

Andreas Malikopoulos

Associate Professor, Department of Mechanical Engineering
University of Delaware, Delaware, USA

Iasson Karafyllis

Associate Professor, School of Applied Mathematical and Physical Sciences
National Technical University of Athens, Athens, Greece

Eleftherios Doitsidis

Assistant Professor, School of Production Engineering and Management
Technical University of Crete, Chania, Greece

Claudio Roncoli

Assistant Professor, Department of Build Environment
Aalto University, Aalto, Finland

ΕΠΤΑΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Τίτλος (ελληνικά/αγγλικά): Επίδραση αυτόματων οχημάτων στη κυκλοφοριακή ροή σε αυτοκινητόδρομους και σηματοδοτούμενες διασταυρώσεις.

Effect of automated vehicles on highway traffic flow and signalized junctions.

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

(Όνοματεπώνυμο διδάκτορα)

Παναγιώτης Τυπάλδος

ΤΡΙΜΕΛΗΣ ΣΥΜΒΟΥΛΕΥΤΙΚΗ ΕΠΙΤΡΟΠΗ:

1. Μάρκος Παπαγεωργίου (Επιβλέπων)
2. Ιωάννης Παπαμιχαήλ
3. Ιωάννης Νικολός

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή την: 22 / 12 / 2022

1. Ιωάννης Παπαμιχαήλ (Καθηγητής) (υπογραφή)

2. Ιωάννης Νικολός (Καθηγητής)

3. Ηλίας Κοσματόπουλος (Καθηγητής, ΔΠΘ)

4. Ανδρέας Μαλικόπουλος (Associate Professor, University of Delaware, ΗΠΑ)

5. Ιάσων Καραφύλλης (Αναπλ. Καθηγητής, ΕΜΠ)

6. Ελευθέριος Δοξοσίδης (Επίκ. Καθηγητής)

7. Claudio Roncoli (Assistant Professor, Aalto University, Φινλανδία)

TECHNICAL UNIVERSITY OF CRETE

Abstract

School of Production, Engineering & Management

Doctor of Philosophy

Effect of Automated Vehicles on Highway Traffic Flow and Signalized Junctions

by Panagiotis TYPALDOS

Over the past years, intensive research has been carried out towards the development of fully automated driving for road vehicles. Fully automated vehicles could improve safety and efficiency of traffic operations by reducing accidents caused by human driver errors; improving driver and passenger comfort; reducing traffic congestion and, at the same time, they contribute to the environmental and economical aspects by reducing fuel consumption and emissions. Although vehicle automation has already led to significant achievements in supporting the driver in various ways, rising the level of automation to fully-automated driving is an extremely challenging problem. This is mainly due to the complexity of real-world environments, including avoidance of static and moving obstacles, compliance with traffic rules and consideration of human driving behavior aspects.

The importance of eco-driving in reducing cumulative fuel consumption of road vehicles is a well-known and widely treated subject. Eco-driving intends to minimize fuel consumption by maneuvering a vehicle with a human or automated driver. In the current work, the eco-driving problem is cast in an optimal control framework with fixed time horizon. For the fuel consumption estimation, a number of alternatives are employed. Initially, a realistic, but nonlinear and non-smooth formula from the literature is considered. Simple smoothing procedures are then applied, so as to enable the application of powerful numerical algorithms for the efficient solution of the resulting nonlinear optimal control problem. Furthermore, suitable quadratic approximations of the nonlinear formula are also considered, which enable analytical problem solutions. A comprehensive comparison on the basis of various driving scenarios demonstrates the features of each alternative approach. In particular, it is demonstrated that the often utilized, but sometimes strongly questioned, square-of-acceleration term delivers excellent approximations for fuel minimizing trajectories in the present setting.

Despite the, seemingly, simple structure of highway environments, human drivers' actions may, often, lead to accidents and excessive fuel consumption. Automated vehicles equipped

by appropriate sensor and path-planning algorithms have the potential to overcome these issues. In this context, a path-planning algorithm for connected and non-connected automated road vehicles on multilane motorways is derived from the opportune formulation of an optimal control problem. In this framework, the objective function to be minimized contains appropriate respective terms to reflect: the goals of vehicle advancement; passenger comfort; and avoidance of collisions with other vehicles and of road departures. Connectivity implies, within the present work, that connected vehicles can exchange with each other (V2V) real-time information about their last generated short-term path. For the numerical solution of the optimal control problem, an efficient feasible direction algorithm (FDA) is used. To ensure high-quality local minima, a simplified Dynamic Programming (DP) algorithm is also conceived to deliver the initial guess trajectory for the start of the FDA iterations. Thanks to very low computation times, the approach is readily executable within a Model Predictive Control (MPC) framework. The proposed MPC-based approach is embedded within a microsimulation platform, which enables the evaluation of a plethora of realistic vehicle driving and advancement scenarios under different vehicles mixes. Results obtained on a multilane motorway stretch indicate higher efficiency of the optimally controlled vehicles in driving closer to their desired speed, compared to ordinary manually driven vehicles. Increased penetration rates of automated vehicles are found to increase the efficiency of the overall traffic flow, benefiting manual vehicles as well. Moreover, connected controlled vehicles appear to be more efficient in achieving their desired speed, compared also to the corresponding non-connected controlled vehicles, due to the improved real-time information and short-term prediction achieved via V2V communication.

Urban areas, and specifically signalized intersections have a crucial role on the safety of drivers, as well as the increased fuel consumption of vehicles. To address these issues, appropriate systems have been developed, known as Green Light Optimal Speed Advisory (GLOSA) systems. In the current work, a discrete-time stochastic optimal control problem has been proposed to address the GLOSA problem in cases where the next signal switching time is decided in real time and is therefore uncertain in advance. The corresponding numerical solution via SDP (Stochastic Dynamic Programming) calls for substantial computational time, which excludes problem solution in the vehicle's on-board computer in real time. To overcome the computation time bottleneck, several modified versions of Dynamic Programming were also developed. As a first attempt, a Discrete Differential Dynamic Programming (DDDP) was employed for the numerical solution of the stochastic optimal control problem. The DDDP algorithm was demonstrated to achieve results equivalent to those obtained with the ordinary SDP algorithm, albeit with significantly reduced computational times. Subsequently, a different modified version of Dynamic Programming, known as Differential Dynamic Programming (DDP) was utilized. For the stochastic GLOSA problem, it is demonstrated that DDP achieves quasi-instantaneous (extremely fast) solutions

in terms of CPU times, which allows for the proposed approach to be readily executable online, within an MPC framework, in the vehicle's on-board computer. The approach is demonstrated by use of realistic examples. It should be noted that DDP does not require discretization of variables, hence the obtained solutions may be slightly superior than the standard SDP solutions. However, in all pre-mentioned GLOSA problems, there was an assumption that the traffic signal is initially red and turns to green, which means that only half traffic light cycle was considered. To this end, the aforementioned problem was extended considering a full traffic light cycle, consisting of four phases: a certain green phase, during which the vehicle can freely pass; an uncertain green phase, in which there is a probability that the traffic light will extend its duration or turn to red at any time; a certain red phase during which the vehicle cannot pass; and an uncertain red phase, in which there is a probability that the red signal may be extended or turn to green at any time. Preliminary results indicate that the proposed SDP approach achieves better average performance, in terms of fuel consumption, compared to the IDM (Intelligent Driver Model) model, which emulates human-driving behavior.

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

Περίληψη

Σχολή Μηχανικών Παραγωγής και Διοίκησης

Doctor of Philosophy

Επίδραση Αυτόματων Οχημάτων στη Κυκλοφοριακή Ροή σε Αυτοκινητόδρομους και
Σηματοδοτούμενες Διασταυρώσεις

Παναγιώτης Τυπάλδος

Τα τελευταία χρόνια, έχει διεξαχθεί εντατική έρευνα για την ανάπτυξη πλήρως αυτοματοποιημένης οδήγησης για οχήματα κινούμενα σε δρόμους. Τα πλήρως αυτόματα οχήματα θα μπορούσαν να βελτιώσουν την ασφάλεια και την αποτελεσματικότητα των κυκλοφοριακών λειτουργιών μειώνοντας τα ατυχήματα που προκαλούνται από ανθρώπινα λάθη, βελτιώνοντας την άνεση του οδηγού και των επιβατών, μειώνοντας την κυκλοφοριακή συμφόρηση και, ταυτόχρονα, συμβάλλοντας σε περιβαλλοντικές και οικονομικές πτυχές μέσω της μείωσης της κατανάλωσης καυσίμων και της εκπομπής ρύπων. Παρόλο που ο αυτοματισμός του οχήματος έχει ήδη οδηγήσει σε σημαντικά επιτεύγματα, υποστηρίζοντας τους οδηγούς με διάφορους τρόπους, η αύξηση του επιπέδου αυτοματισμού σε πλήρως αυτοματοποιημένη οδήγηση είναι ένα εξαιρετικά δύσκολο πρόβλημα. Αυτό οφείλεται κυρίως στην πολυπλοκότητα των περιβάλλοντων του φυσικού κόσμου, συμπεριλαμβανομένης της αποφυγής στατικών και κινούμενων εμποδίων, της συμμόρφωσης με τους κανόνες οδικής κυκλοφορίας και της ανθρώπινης οδηγικής συμπεριφοράς.

Η σημασία της οικολογικής οδήγησης για τη μείωση της κατανάλωσης καυσίμου των οχημάτων στου δρόμους είναι ένα πολύ γνωστό και ευρέως αντιμετωπισμένο θέμα. Η οικολογική οδήγηση σκοπεύει στο να ελαχιστοποιήσει την κατανάλωση καυσίμων μέσω κατάλληλων ελιγμών ενός αυτόματου ή μη οχήματος. Στην τρέχουσα εργασία, το πρόβλημα της οικολογικής οδήγησης προσεγγίζεται ως ένα πρόβλημα βέλτιστου ελέγχου με σταθερό χρονικό ορίζοντα. Για την εκτίμηση της κατανάλωσης καυσίμου, χρησιμοποιούνται διάφορες εναλλακτικές λύσεις. Αρχικά, εξετάζεται ένα ρεαλιστικό, αλλά μη-γραμμικό και μη-λείο μοντέλο κατανάλωσης καυσίμων από τη βιβλιογραφία. Στη συνέχεια εφαρμόζονται απλές διαδικασίες εξομάλυνσης, ώστε να καταστεί δυνατή η εφαρμογή αριθμητικών αλγορίθμων για την αποτελεσματική λύση του προκύπτοντος

μη-γραμμικού προβλήματος βέλτιστου ελέγχου. Επιπλέον, εξετάζονται κατάλληλες τετραγωνικές προσεγγίσεις του μη-γραμμικού μοντέλου, οι οποίες επιτρέπουν τη αναλυτική λύση του προβλήματος. Μια εκτενής σύγκριση, με βάση διάφορα σενάρια οδήγησης, δείχνει τα χαρακτηριστικά κάθε εναλλακτικής προσέγγισης. Συγκεκριμένα, αποδεικνύεται ότι ο συχνά χρησιμοποιούμενος, αλλά έντονα αμφισβητούμενος, τετραγωνικός όρος της επιτάχυνσης παρέχει εξαιρετικές προσεγγίσεις για τροχιές που ελαχιστοποιούν την κατανάλωση καυσίμων.

Παρά την, φαινομενικά, απλή δομή του περιβάλλοντος των αυτοκινητοδρόμων, οι ενέργειες των ανθρώπινων οδηγών μπορεί, συχνά, να οδηγήσουν σε ατυχήματα και αυξημένη κατανάλωση καυσίμων. Τα αυτόματα οχήματα εξοπλισμένα με κατάλληλους αισθητήρες και αλγόριθμους προγραμματισμού πορείας έχουν τη δυνατότητα να αντιμετωπίσουν αυτά τα ζητήματα. Σε αυτό το πλαίσιο, ένας αλγόριθμος προγραμματισμού πορείας, για συνδεδεμένα και μη-συνδεδεμένα αυτόματα οχήματα σε αυτοκινητόδρομους πολλαπλών λωρίδων, εξάγεται από την κατάλληλη διαμόρφωση ενός προβλήματος βέλτιστου ελέγχου. Σε αυτό το πλαίσιο, η αντικειμενική συνάρτηση προς ελαχιστοποίηση περιέχει κατάλληλους όρους που αντικατοπτρίζουν: τους στόχους της κίνησης του οχήματος, την άνεση των επιβατών, την αποφυγή συγκρούσεων με άλλα οχήματα και την αποφυγή κίνησης εκτός δρόμου. Η συνδεσιμότητα συνεπάγεται, στην παρούσα εργασία, ότι τα συνδεδεμένα οχήματα μπορούν να ανταλλάσσουν μεταξύ τους (V2V) πληροφορίες σε πραγματικό χρόνο σχετικά με την τελευταία βραχυπρόθεσμη διαδρομή τους. Για την αριθμητική επίλυση του προβλήματος βέλτιστου ελέγχου, χρησιμοποιείται ένας αποτελεσματικός αλγόριθμος εφικτών κατευθύνσης. Για να εξασφαλιστούν τοπικά ελάχιστα υψηλής ποιότητας, σχεδιάστηκε επίσης ένας απλοποιημένος αλγόριθμος Δυναμικού Προγραμματισμού, για να υπολογίσει μια αποτελεσματική αρχική τροχιά για τον αλγόριθμο εφικτών κατευθύνσεων. Χάρη στους πολύ χαμηλούς υπολογιστικούς χρόνους, η προσέγγιση είναι εκτελέσιμη σε πραγματικό χρόνο στο πλαίσιο ελέγχου προβλεπτικού μοντέλου (MPC) που αντιμετωπίζει διάφορες εμπλεκόμενες αβεβαιότητες. Η προτεινόμενη προσέγγιση ενσωματώνεται σε μια πλατφόρμα μικροσκοπικής προσομοίωσης, η οποία επιτρέπει την αξιολόγηση πληθώρας ρεαλιστικών σεναρίων κίνησης οχημάτων κάτω από διαφορετικά μείγματα οχημάτων. Τα αποτελέσματα που προέκυψαν, σε έναν ευθύγραμμο αυτοκινητόδρομο πολλαπλών λωρίδων, υποδεικνύουν υψηλότερη αποτελεσματικότητα των βέλτιστα ελεγχόμενων οχημάτων στο να διατηρηθούν κοντά στην επιθυμητή τους ταχύτητα, σε σύγκριση με τα χειροκίνητα οχήματα. Η αύξηση του ποσοστού των αυτόματων οχημάτων διαπιστώθηκε ότι οδηγεί σε συνολικά αποτελεσματικότερη κυκλοφοριακή ροή, ωφελώντας και τα χειροκίνητα οχήματα. Επιπλέον, τα συνδεδεμένα ελεγχόμενα οχήματα φαίνεται να είναι πιο αποτελεσματικά στην επίτευξη της επιθυμητής ταχύτητάς τους, σε σύγκριση επίσης με τα αντίστοιχα μη-συνδεδεμένα ελεγχόμενα οχήματα, λόγω της βελτιωμένης

πληροφόρησης σε πραγματικό χρόνο και της βραχυπρόθεσμης πρόβλεψης που επιτυγχάνονται μέσω της επικοινωνίας V2V.

Οι αστικές περιοχές και συγκεκριμένα οι σηματοδοτούμενες διασταυρώσεις έχουν καθοριστικό ρόλο στην ασφάλεια των οδηγών, καθώς και στην αυξημένη κατανάλωση καυσίμων των οχημάτων. Για την αντιμετώπιση αυτού του ζητήματος, έχουν αναπτυχθεί κατάλληλα συστήματα, γνωστά ως συστήματα GLOSA (Green Light Optimal Speed Advisory). Στην τρέχουσα εργασία, έχει προταθεί ένα πρόβλημα στοχαστικού βέλτιστου ελέγχου διακριτού χρόνου για την αντιμετώπιση του προβλήματος GLOSA σε περιπτώσεις όπου ο χρόνος της επόμενης εναλλαγής σήματος αποφασίζεται σε πραγματικό χρόνο και επομένως είναι αβέβαιος εκ των προτέρων. Η αντίστοιχη αριθμητική λύση μέσω ΣΔΠ (Στοχαστικού Δυναμικού Προγραμματισμού) απαιτεί αυξημένο υπολογιστικό χρόνο, ο οποίος αποκλείει τη λύση του προβλήματος στον υπολογιστή του οχήματος σε πραγματικό χρόνο. Για την αντιμετώπιση του αυξημένου υπολογιστικού φόρτου, αναπτύχθηκαν επίσης αρκετές τροποποιημένες εκδοχές του Δυναμικού Προγραμματισμού. Ως πρώτη προσπάθεια, χρησιμοποιήθηκε Διακριτικός Διαφορικός Δυναμικός Προγραμματισμός (ΔΔΔΠ) για την αριθμητική επίλυση του προβλήματος του στοχαστικού βέλτιστου ελέγχου. Ο αλγόριθμος ΔΔΔΠ αποδείχθηκε ότι επιτυγχάνει αποτελέσματα ισοδύναμα με αυτά που λαμβάνονται με τον συνηθισμένο αλγόριθμο ΣΔΠ, αλλά με σημαντικά μειωμένους υπολογιστικούς χρόνους. Στη συνέχεια, χρησιμοποιήθηκε μια διαφορετική τροποποιημένη έκδοση του Δυναμικού Προγραμματισμού, γνωστή ως Διαφορικός Δυναμικός Προγραμματισμός (ΔΔΠ). Για το στοχαστικό πρόβλημα GLOSA, αποδεικνύεται ότι ο ΔΔΠ επιτυγχάνει σχεδόν στιγμιαίες (εξαιρετικά γρήγορες) λύσεις όσον αφορά τους υπολογιστικούς χρόνους, γεγονός που επιτρέπει την εύκολη εκτέλεση της προτεινόμενης προσέγγισης σε πραγματικό χρόνο στο πλαίσιο ελέγχου προβλεπτικού μοντέλου, στον υπολογιστή του οχήματος. Θα πρέπει να σημειωθεί ότι ο ΔΔΠ δεν απαιτεί διακριτοποίηση των μεταβλητών, επομένως οι λύσεις που λαμβάνονται μπορεί να είναι ελαφρώς καλύτερες από τις τυπικές λύσεις του ΣΔΠ. Ωστόσο, σε όλα τα προαναφερθέντα προβλήματα GLOSA, υπήρχε η υπόθεση ότι το σήμα κυκλοφορίας είναι αρχικά κόκκινο και γίνεται πράσινο, πράγμα που σημαίνει ότι λήφθηκε υπόψη μόνο μισός κύκλος ενός φωτεινού σηματοδότη. Για το σκοπό αυτό, το προαναφερθέν πρόβλημα επεκτάθηκε λαμβάνοντας υπόψη έναν πλήρη κύκλο φωτεινών σηματοδοτών, που αποτελείται από τέσσερις φάσεις: μια ορισμένη πράσινη φάση, κατά την οποία το όχημα μπορεί να περάσει ελεύθερα, μια αβέβαιη πράσινη φάση, στην οποία υπάρχει πιθανότητα το φανάρι να παρατείνει τη διάρκειά του ή να γίνει κόκκινο ανά πάσα στιγμή, μια ορισμένη κόκκινη φάση κατά την οποία το όχημα δεν μπορεί να περάσει, και μια αβέβαιη κόκκινη φάση, στην οποία υπάρχει πιθανότητα το κόκκινο σήμα να παραταθεί ή να γίνει πράσινο ανά πάσα στιγμή. Τα αποτελέσματα υποδεικνύουν ότι η προτεινόμενη προσέγγιση ΣΔΠ επιτυγχάνει καλύτερη μέση απόδοση, όσον αφορά την κατανάλωση καυσίμων, σε

σύγκριση με το μοντέλο **IDM (Intelligent Driver Model)**, το οποίο μιμείται την ανθρώπινη συμπεριφορά οδήγησης.

Acknowledgements

The path towards a PhD is very difficult, but also full of fulfilling moments. Having reached the end of this path, I would like to thank and express my gratitude to all the people who have accompanied me along the way.

I would like to start with my advisors Prof. Papageorgiou and Prof. Papamichail. I am very grateful and fortunate to have these two excellent advisors. Their academic excellence and constructive guidance, as well as their continuous support and patience, have been the most valuable ingredients for both my scientific work and my development as a researcher and as a person.

I would also like to thank all the committee members, Prof. Nikolos, Prof. Kosmatopoulos, Prof. Malikopoulos, Prof. Karafyllis, Prof. Doitsidis, and Prof. Roncoli, for accepting to participate in the defense of my dissertation and for their fruitful and insightful comments on various parts of my dissertation. Of course, I would also like to thank all members and colleagues of the Dynamic Systems and Simulation Laboratory (DSSL) for their support and for the many interesting discussions over several topics and issues during this time.

Last but not least, I would like to thank my family for their endless support and encouragement. Their unconditional love is always the best motivation and encouragement in my life.

This research received funding partially from the European Commission under the European Union's Seventh Framework Programme (FP/2007-2013) / FP7-ICT-2013.3.4, project Local4Global (n. 611538); partially from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. [321132], project TRAMAN21 and; partially from the European Research Council under the European Union's Horizon 2020 Research and Innovation programme / ERC Grant Agreement n. [833915], project TrafficFluid.



Contents

Abstract	iii
Περίληψη	vi
Acknowledgements	x
Contents	xi
List of Figures	xv
List of Tables	xviii
1 Introduction and State of the Art	1
1.1 Introduction	1
1.1.1 Motivations	1
1.1.2 Outline and Contributions	3
1.2 State of the Art	6
1.2.1 Eco-Driving as an Optimization Problem	6
1.2.2 Path-Planning for Automated vehicles	7
1.2.3 Green Light Optimal Speed Advisory (GLOSA) Systems	10
2 Minimization of Fuel Consumption for Vehicle Trajectories	13
2.1 Abstract	13
2.2 Introduction	13
2.3 Fuel Consumption Criterion	16
2.3.1 Energy-Related Model for Estimating Fuel Consumption	16
2.3.2 Smooth Form of the ARRB Fuel Consumption Function	18
2.3.3 Quadratic Approximation of the Fuel Consumption Model	19
2.4 Optimal Control Problems	20

2.4.1	Approach 1: Analytic Solution of Approximated Fuel Consumption Model	21
2.4.1.1	Optimal Control Methodology for the Approximated Fuel Consumption Model	21
2.4.1.2	Approach 1-a: A Special Case of the Approximated Fuel Consumption Model	23
2.4.2	Approach 2: Numerical Solution of the Smoothed Fuel Consumption Model	23
2.4.2.1	Feasible Direction Algorithm	24
2.4.2.2	Discrete-Time Optimal Control Problem Formulation	26
2.4.3	A Stop Point Scenario	27
2.4.3.1	Continuous-Time Formulation	27
2.4.3.2	Discrete-Time Formulation	29
2.5	Results and Comparisons	29
2.5.1	Solution Approaches 1 and 1-a for Case 1	30
2.5.2	Comparison of the Search Direction Algorithms for Approach 2 for Case 1	31
2.5.3	Comparison of Approaches 1, 1-a and 2 for Case 1	32
2.5.4	Approaches 1-a and 2 for Different Time Horizons for Case 1	33
2.5.5	Approaches 1-a and 2 for Case 2 with Different Stop Point Values	34
2.5.6	Fuel Consumption Maximization Problem for Cases 1 and 2	36
2.6	Conclusion	39
3	Optimization-Based Path-Planning for Connected and non-Connected Automated Vehicles	40
3.1	Abstract	40
3.2	Introduction	41
3.3	Optimal Control Problem Formulation	45
3.3.1	Problem Variables, State-Equations and Constraints	46
3.3.2	Optimization Objectives	48
3.3.2.1	Objective Function	48
3.3.2.2	Road Boundaries Term	49
3.3.2.3	Collision Avoidance Term	49
3.3.3	Problem Formulation	52
3.4	Numerical Solution and Model Predictive Control	53
3.4.1	Feasible Direction Algorithm (FDA)	53
3.4.2	Dynamic Programming (DP)	55
3.4.3	Safety Override	59

3.4.4	Model Predictive Control	60
3.5	Simulation Testing and Results	62
3.5.1	Simulation Environment	62
3.5.2	Results	65
3.6	Conclusions	70
4	Vehicle Trajectory Specification in Presence of Traffic Lights with Known or Uncertain Switching Times	73
4.1	Abstract	73
4.2	Introduction	74
4.3	Optimal Control Problem with Known Signal Switching Time	77
4.3.1	Problem Formulation	77
4.3.2	Analytical Solution	79
4.4	Stochastic Optimal Control Problem with Uncertain Signal Switching Time	82
4.4.1	Problem Variables and State Equations	83
4.4.2	Objective Criterion	84
4.4.3	Numerical Solution Algorithm	85
4.4.4	Discussion	88
4.5	Results	89
4.5.1	Investigated Scenarios	89
4.5.2	Deterministic GLOSA	89
4.5.3	Stochastic GLOSA	90
4.6	Conclusion	95
5	Modified Dynamic Programming Algorithms for GLOSA Systems with Stochastic Signal Switching Times	96
5.1	Abstract	96
5.2	Introduction	97
5.3	Optimal Control-Based GLOSA with Known or Unknown Signal Switching Time	100
5.3.1	Known Signal Switching Time - Deterministic GLOSA	100
5.3.2	Uncertain Signal Switching Time Problem – Stochastic GLOSA	101
5.3.3	Discrete SDP Numerical Solution Algorithm	104
5.4	Discrete Differential Dynamic Programming (DDDP)	106
5.5	Differential Dynamic Programming (DDP)	107
5.5.1	General Constrained DDP	107
5.5.2	Constrained DDP for the GLOSA Problem	110
5.6	Demonstration Results and Comparison	112
5.6.1	Scenarios Setup	112

5.6.2	DDDP Algorithm Results	113
5.6.3	DDP Algorithm Results	116
5.6.4	Summarized Results	116
5.7	Conclusions	118
6	GLOSA System with Uncertain Green and Red Signal Phases	124
6.1	Abstract	124
6.2	Introduction	124
6.3	Problem Formulation and Solution	126
6.4	Preliminary Results and Discussion	130
6.5	Conclusions	131
7	Conclusion	133
	Bibliography	136

List of Figures

2.1	The ARRB Fuel Consumption function and its smoothed variants versus the acceleration a (assuming the value of speed is constant, $v = 15$ m/s) for different values of α_s	19
2.2	A vehicle starting from a given initial state (x_0, v_0) , aims to reach a final state (x_e, v_e) with the presence of a traffic light at a given position that switches to green at time t_r	28
2.3	Graphical representation of the solution trajectories, for the Approach 1 (blue lines), Approach 1-a (red lines) and Approach 2 (magenta lines), for (a) Scenario 1 and (b) Scenario 2.	31
2.4	Control and states for Approach 1-a (red lines) and Approach 2 (blue lines) for the cases of (a) $T = 19$ s and $K = 100$, (b) $T = 38$ s and $K = 200$, (c) $T = 57$ s and $K = 300$	35
2.5	Graphical representation of the solution trajectories, for different values of k_1 , where the blue lines are for Case 1 and the red, magenta, green, black lines for Case 2 with $k_1 = 60, 70, 80, 90$, respectively, for (a) Approach 1-a and (b) Approach 2.	36
2.6	Comparison between minimization (red line) and maximization (blue line) problems of the smoothed fuel consumption model, for (a) Case 1 for Scenario 1 and (b) Case 2 for $k_1 = 70$	38
3.1	The penalty function of road boundaries.	49
3.2	Illustration of the collision avoidance term (orange ellipsoid), $c_i(x, y)$, for different ego vehicle speeds with time gap $\omega = 1.2$ s and obstacle speed $v_i = 20$ m/s, compared to the corresponding space gap of the constant time gap policy (red rectangle). The lateral dashed line indicates the physical vehicle centre location.	51
3.3	AV lateral position trajectory (blue line) and its mapping into Aimsun lanes (orange lines).	64

3.4	Trajectories of several representative AVs during the simulation, displaying longitudinal and lateral speeds (orange line is the longitudinal desired speed) and accelerations, longitudinal jerk, and lateral continuous position produced by FDA (blue lines), along with the corresponding mapping on lanes (orange lines).	66
3.5	Average delay time, speed, deviation from target speed and number of lane changes for: i) connected (green lines); and ii) non-connected (blue lines) AVs; for a demand of 3.000 veh/h. Solid lines represent the average of the whole section, while dense-dashed and sparse-dashed lines reflect on average results for automated and manually driven vehicles, respectively.	67
3.6	Average delay time, speed, deviation from target speed and number of lane changes for: i) connected (green lines); and ii) non-connected (blue lines) AVs; for a demand of 5.000 veh/h. Solid lines represent the average of the whole section, while dense-dashed and sparse-dashed lines reflect on average results for automated and manually driven vehicles, respectively.	68
3.7	Average number of (re-)plans for different penetration rates of the two evaluated cases: non-connected AVs (blue); and connected AVs (green) and for different demand levels: a) 3.000 and b) 5.000 veh/h	70
4.1	Optimal final time t_e (blue dotted marks) and acceleration cost (red star marks) versus the weight w	79
4.2	Illustration of the 3-D grid of time and state space.	87
4.3	Graphical representation of the optimal trajectories for: (a) Scenario 1 and (b) Scenario 2 (blue lines), along with the corresponding trajectories of the UP solution (red dotted lines). Also, the traffic light phases are represented as thick red and green lines for red and green phases, respectively.	90
4.4	Cost (red dotted line) and execution time (blue line) versus discretization interval Δ	91
4.5	Optimal states and control trajectories of SDP (blue lines) compared with those of the sub-optimal approach (red lines) for: (a) Scenario 2 and (b) Scenario 2a. The traffic light phases are represented as a straight line which is red for the red phase, yellow for the stochastic switching period, and green for the rest. The bottom-right diagram in each scenario displays the resulting square-of-acceleration cost (blue lines) and the fuel consumption (red lines), the latter based on the Australian Road Research Board (ARRB) model, for each approach (continuous lines for the SDP and dashed lines for the sub-optimal approach) versus the actual switching time in the range $[k_{\min}, k_{\max}]$	93

4.6	Average percentage difference of fuel consumption of SDP versus the sub-optimal approach, calculated with the ARRB fuel consumption model, for different initial state values x_0 (in m), v_0 (in m/s) for: (a) the setup of Scenario 2 and (b) same setup but different time-window of switching times ($k_{\min}, k_{\max} = [40, 60]$) with extended time horizon and distance from the traffic signal ($x_{\max} = 300$).	94
5.1	A vehicle starting from a given initial state (x_0, v_0) aims to reach a fixed final state (x_e, v_e) in presence of a traffic signal at a given position x_1 that switches to green at time t_1	100
5.2	Received (blue dashed line) and optimal (orange line) (a) acceleration and (b) speed trajectories of DDDP algorithm in each iteration for Scenario 1.	114
5.3	Received (blue dashed line) and computed (orange line) trajectories of (a) acceleration and (b) speed at each iteration of DDP algorithm for Scenario 1.	117
5.4	Optimal acceleration and speed trajectories (orange lines) of DDP, compared with the corresponding optimal trajectories of the one-shot SDP (blue dashed lines) of (a) Scenario 1, (b) Scenario 2 and (c) Scenario 3.	118
5.5	Computation times until convergence of: (a) DDDP and (b) DDP algorithms for different initial conditions.	119
5.6	Received (blue dashed line) and optimal (orange line) (a) acceleration and (b) speed trajectories of DDDP algorithm in each iteration for Scenario 2.	120
5.7	Received (blue dashed line) and optimal (orange line) (a) acceleration and (b) speed trajectories of DDDP algorithm in each iteration for Scenario 3.	121
5.8	Received (blue dashed line) and computed (orange line) trajectories of (a) acceleration and (b) speed at each iteration of DDP algorithm for Scenario 2.	122
5.9	Received (blue dashed line) and computed (orange line) trajectories of (a) acceleration and (b) speed at each iteration of DDP algorithm for Scenario 3.	122
6.1	Traffic light phases.	127
6.2	Optimal state and control trajectories of SDP (blue lines) versus IDM (magenta lines). In (a) and (d), the actual switching times are indicated with vertical dashed lines for Scenarios 1 and 2, respectively.	131

List of Tables

2.1	Comparison of search direction methods of Approach 2 for (a) Scenario 1 and (b) Scenario 2	33
2.2	Numerical solutions of Approaches 2 and 1-a for different values of k_1 , using the DFP algorithm. The Cost is the fuel consumption based on the ARRB model.	37
3.1	Average execution times of the three proposed algorithms, FDA, DP and Forward DP, for different planning horizons.	70
5.1	Investigated scenarios.	112
5.2	Optimal cost evolution and discretization change in each iteration of DDDP algorithm.	115
5.3	Comparison of iterations, CPU times until convergence and obtained optimal cost of SDP, DDDP and DDP	117
5.4	Performance of DDDP algorithm in terms of CPU-time and optimal cost for different initial values of \mathbf{C} and Δa	123

Chapter 1

Introduction and State of the Art

1.1 Introduction

1.1.1 Motivations

In an era of climate crisis and cheap energy shortage, actions toward environment protection should be taken across all technical fields. The impact of transportation systems on both environmental pollution and energy resources shortage is significant as the current trend towards increased mobility seems to run counter to the resolve of these issues, thus approaches that lead to increased fuel efficiency are crucial.

Regarding road traffic transportation, several studies note the alarming economical and environmental impacts as well as the affect on road traffic conditions. In (Schrank et al., 2011), it is reported that, in 2020, the amount of fuel consumed and gas emissions in USA were estimated to 1.7 billion gallons and 18 million tons, respectively. Similarly, the travel delay was estimated to 4.3 billion hours, with the congestion cost being 101 billion dollars. Additionally, the demand for transportation is expected to increase, especially in the cities around the world, as the number of population increases rapidly. To this end, benefits of fuel efficiency in road vehicles are twofold, as it is directly correlated with reduction of fuel emissions, as well as to economic aspects, as fuel economy means less expenses for the driver. Indeed, a wide range of technologies have been developed in recent years to decrease fuel consumption of vehicles, including efficient engines, adjusted vehicle designs and lighter chassis. Furthermore, considerable efforts in the development and deployment of efficient intelligent transportation systems (including real-time traffic signals) lead to reduced congestion and fuel consumption.

In the past decade, automated driving has attracted strong interest in industry and scientific community. This is fostered by strong technological advancements, compared to which human driving capabilities appear limited in terms of perception of the driving environment, reaction time, and real-time decision efficiency. Automated vehicles (AVs) are expected to revolutionize the transportation sector, improving on traffic safety and efficiency by

overcoming the limitations of human driving. Sensors and path planning algorithms play a significant role on the path from human driving towards full vehicle automation (Marti et al., 2019). The current driving pattern followed by human drivers is resulting in underuse of the road infrastructure and excessive fuel consumption due to non-optimal accelerations. Human perception of the surroundings is limited compared to the wide range of sensors that are available onboard automated vehicles. According to statistics (Singh, 2015), the reasons for around 94% of the road accidents are attributed to drivers, and related driving errors can be classified to be due to perception, recognition, decision, performance, and non-performance. Each year, road accidents result in approximately 1.35 million fatalities and leave some 50 million of injured or disabled worldwide. Road congestion is another major issue, causing excessive delays, fuel consumption and emissions around the globe (Goniewicz et al., 2016; Montanaro et al., 2019). Since the introduction of motorized vehicles, an extensive amount of research has been going on to make the roads safer and more efficient. However, accidents and congestion are still regular plaguing occurrences across the world, calling for comprehensive and radical solutions. One potential solution would be introducing automated vehicles on the roads with appropriate movement strategies. Automation is based on high-level precision while performing ordinary tasks, that are boring and distraction-prone for humans. However, due to the strongly dynamic nature of road traffic in both longitudinal and lateral directions, complex algorithms, like optimal model predictive control, nonlinear feedback control, reinforcement learning, and artificial intelligence, are necessary (González et al., 2015; Karafyllis et al., 2021; Troullinos et al., 2021; Yanumula et al., 2021). With appropriate sharing of information via Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication, automated vehicles may be enabled to drastically alleviate the problems of safety and congestion (Diakaki et al., 2015; Loke, 2019). Traffic lanes were introduced to aid the human driving, as they minimize the need for lateral and rear monitoring; albeit with an exception during lane-changing, which calls for simultaneous multi-direction monitoring, prediction of multiple vehicle movements and fast decision making. Around 10% of all road accidents occur while performing the lane-change by human drivers (Rahman et al., 2013). Besides, due to diverse vehicle dimensions, lanes are designed to fit the widest vehicles in regular traffic, which are big trucks. Thus, a great deal of lateral space is left unused when smaller vehicles are using the same lane structure. At traffic level, lane-changes lead to capacity losses and are known to trigger traffic breakdown at critical traffic conditions.

Traffic signals guarantee, in the first place, the safe crossing of vehicles at urban junctions in cities around the world. Clearly, enforcing safety via traffic lights implies that some vehicles will have to stop in front of a red light; and then accelerate after the traffic light switching to green; something that affects the fuel consumption of concerned vehicles. To reduce the resulting vehicle delays and number of stops, several algorithms have been proposed and deployed over the past decades, aiming at optimizing the traffic signals

operation, see (Hounsell, McDonald, 2001; Papageorgiou et al., 2003) for overviews. Fixed-time signal plans are derived off-line for respective times-of-day (e.g., morning peak hour, off-peak etc.) by use of appropriate optimization codes, based on historical constant demands; and are applied without deviations; this implies that switching times of the traffic lights are always known in advance. In contrast, real-time (or traffic-responsive or adaptive) signal control strategies make use of real-time measurements to calculate in real time suitable signal settings. Depending on the employed signal control strategy, the control update period may range from one second to one signal cycle; clearly, for real-time signals, the next switching time is not known before the switching decision is actually made. It should be noted that real-time strategies are deemed to be more advanced and potentially more efficient than fixed signal operation. Fuel consumption is increasingly considered as an optimization or evaluation criterion while developing and deploying signal control systems (Jamshidnejad et al., 2017).

Although traffic signals greatly contribute to the safety of the vehicle drivers, they are also one of the major reasons of decreased traffic flow and, consequently, of increased emissions and fuel consumption. Human drivers tend to maintain constant speed while approaching a traffic signal, which leads to excessive deceleration in order to fully stop and, apparently, acceleration when the traffic signal switches to green. On the other hand, by appropriately adjusting the speed either by accelerating or decelerating, the vehicle may be able to cross the traffic signal after it has switched to green, or smoothly stop in case of red, in which cases no fuel-intensive acceleration needs to be applied. This dilemma of vehicle movement towards a red traffic signal is addressed by use appropriately designed systems called Green Light Optimal Speed Advisory (GLOSA) systems (Stahlmann et al., 2016). GLOSA systems, by receiving information from the traffic signal about the potential duration of green or red phase duration, guide the driver (or automated vehicle) by giving speed advice which ensures that, by the time the vehicle reaches the traffic signal, it will be green; or, more specifically, that the fuel consumption and emissions will be minimized.

1.1.2 Outline and Contributions

Following the introduction, the outline of the thesis is presented in this section. In section 1.2, a literature review, regarding different applications of trajectory planning methodologies for automated vehicles, is presented.

In section 2, the minimization of a vehicle's fuel consumption by optimizing its kinematic trajectories, while driving from a given initial state to a given final state within a fixed time horizon, is considered. The minimization problem is formulated as an optimal control problem, which minimizes a specific energy-related, realistic but complex fuel consumption function. Section 2.3 presents the fuel consumption model used as cost criterion. Specifically,

Section 2.3.1 describes, in details, the complex fuel-consumption model, while in Sections 2.3.2 and 2.3.3 a smooth form, as well as a quadratic approximation are derived. Section 2.4 contains the formulation of the optimal control problem, along with the analytical solution of the approximated fuel model, followed by the numerical solution of the smoothed cost criterion. In Section 2.5, the results of the approximate and exact solutions are compared for a number of driving scenarios, and it is demonstrated that simpler (quadratic) functions may indeed serve as appropriation tools for obtaining good approximations of eco-driving solutions.

In section 3, a path-planning algorithm for connected and non-connected automated road vehicles on multilane motorways is derived from the opportune formulation of an optimal control problem. Section 3.3 presents the formulation of the problem, starting with the dynamics and the constraints of each AV in Section 3.3.1. The objective function with all the contained terms are thoroughly explained in Section 3.3.2, with the corresponding terms reflecting the goals of vehicle advancement; passenger comfort; and avoidance of collisions with other vehicles and of road departures. Section 3.4 includes an efficient feasible direction algorithm (FDA) and a simplified Dynamic Programming (DP) algorithm that utilized for the numerical solution of the optimal control problem. Note that, the combination of the two algorithms ensure high-quality local minima, as the DP algorithm is conceived to deliver good initial guess trajectory for the start of the FDA iterations. The proposed approach features very low computation times, which enables the open-loop solution procedure to be case in a model predictive control (MPC) framework as described in Section 3.4.4. Section 3.5 contains the simulation results obtained on a multilane motorway stretch for different penetration rates of connected and non-connected automated vehicles. The results indicate indicate higher efficiency of the optimally controlled vehicles in driving closer to their desired speed, compared to ordinary manually driven vehicles, while Increased penetration rates of automated vehicles are found to increase the efficiency of the overall traffic flow.

Chapters 4 address the problem of generating optimal trajectories for vehicles crossing a signalized junction, with traffic signals operated in either fixed-time or real-time (adaptive) mode. For the first case, in Section 4.3, the problem is formulated as an optimal control problem with pertinent final state for the vehicle and is solved analytically via PMP. Section 4.4 presents the case where the traffic light's switching time is stochastic, with known probability distribution, is addressed, and the problem is cast in the format of a stochastic optimal control problem, which is solved numerically using SDP techniques. In contrast to other works from the literature, the stochastic optimal control problem is designed to extend up to the vehicle's final state by exploiting appropriately the analytic solution of the fixed switching time problem derived earlier, something that guarantees that the obtained solution is consistent, independently of the actual switching time occurrence. The proposed SDP algorithm may take several minutes to execute, which implies that the solution is not real-time

feasible and can therefore not be obtained on-board the vehicle, but must be executed offline, at the infrastructure side, and be communicated to approaching vehicles according to their current state. To substantially reduce the computation time and memory requirements for the solution of the stochastic GLOSA problem and enable its solution on-board the vehicle, two modified (S)DP algorithms are developed in Chapter 5. The first modified algorithm, known as, Discrete Differential Dynamic Programming (DDDP) algorithm is described in section 5.4. DDDP is an iterative algorithm, which solves at each iteration the conventional DP problem but in strongly reduced state space around an initial, given, trajectory. Thus, the computational and memory requirements are lower compared to the one-shot full problem solution, leading in very fast solutions (around 1 s CPU time). As an attempt to further reduce the computation time for the solution of the stochastic GLOSA problem and enable its solution on vehicle's side, a Differential Dynamic Programming (DDP) algorithm is utilized in section 5.5. DDP is found to deliver solutions quasi-instantaneously, which enables the approach to be readily implementable in an MPC (model predictive control) framework and be executed online on the vehicle's computer. In the aforementioned GLOSA problem, there is the assumption that the traffic signal is initially red and turns to green, which means that only half traffic light cycle was considered. To this end, an extension of the stochastic optimal control problem is described in Section 6, where a full traffic light cycle is considered, consisting of four phases: a certain green phase, during which the vehicle can freely pass; an uncertain green phase, in which there is a probability that the traffic light will extend its duration or turn to red at any time; a certain red phase during which the vehicle cannot pass; and an uncertain red phase, in which there is a probability that the red signal may be extended or turn to green at any time.

The list of the publications included in this thesis is given below :

- Chapter 2: P. Typaldos et al. (2020b). “Minimization of fuel consumption for vehicle trajectories”. *IEEE Transactions on Intelligent Transportation Systems* 21.4, pp. 1716–1727
- Chapter 3: P. Typaldos et al. (2022b). “Optimization-based path-planning for connected and non-connected automated vehicles”. *Transportation Research Part C: Emerging Technologies* 134, p. 103487
- Chapter 4: P. Typaldos et al. (2020a). “Vehicle trajectory specification in presence of traffic lights with known or uncertain switching times”. *Transportation research record* 2674.8, pp. 53–66
- Chapter 5: P. Typaldos, M. Papageorgiou (2022). “Modified dynamic programming algorithms for GLOSA systems with stochastic signal switching times”. URL: <https://arxiv.org/abs/2211.12159> (submitted)

- Chapter 6: P. Typaldos et al. (2022a). “GLOSA system with uncertain green and red signal phases”. *4th Symposium on Management of Future motorway and urban Traffic Systems (MFTS 2022)*

1.2 State of the Art

1.2.1 Eco-Driving as an Optimization Problem

Research related to vehicle control for improving its fuel economy has a long track. Some early studies were conducted to determine the optimal cruising velocity of a vehicle based on its operating characteristics (Gilbert, 1976). In other studies, such as (Hooker, 1988), the optimal values of acceleration for starting up and cruising speed over hilly terrains or flat roads are determined for fuel-efficient driving. The problem of improving fuel economy of road vehicles has been recently often referred to as eco-driving (Barkenbus, 2010). Ecological (eco) driving is a way of maneuvering a vehicle with a human (or, more recently, an automated) driver that is intended to minimize fuel consumption by trading off the most efficient driving behaviour of the vehicle. Aside from various physical factors, driving style has a great influence on vehicle emissions and energy consumption (Van Mierlo et al., 2004). Several studies have indicated that eco-driving can improve fuel economy by 15-25% (Kamal et al., 2011).

The eco-driving problem can be formulated as an optimal control problem. To solve the problem, one may employ Dynamic Programming (DP) techniques (Hooker, 1988; Mensing et al., 2011; Monastyrsky, Golownykh, 1993; Ozatay et al., 2014); however, DP application may face real-time feasibility difficulties, in cases where online solutions are required, due to the method's notorious "curse of dimensionality". Alternatively, one may transform the optimal control problem (in a number of possible ways) to a Nonlinear Programming (NLP) problem that can be solved using numerical NLP methods, see e.g. (Saboochi, Farzaneh, 2009; Xu et al., 2014); note, however, that even some NLP-based approaches may lead to computational times that are not feasible in real-time applications. Another approach for the solution of the fuel/energy efficiency problem is to use Pontryagin's Minimum Principle (PMP), where the optimal control problem is transformed to a boundary value problem (Petit, Sciarretta, 2011; Schwarzkopf, Leipnik, 1977). In (Ozatay et al., 2012), the authors develop an analytical solution to the optimization problem of finding the minimum fuel consumption of a vehicle given the road and possibly the traffic light information. Also, the authors in (Kim et al., 2010) proposed an optimal control based on PMP approach. The simulation results are the same as the global optimal control based on dynamic programming results. The work in (Lawitzky et al., 2013) concentrates on energy optimal approaching red traffic lights. The energy efficiency optimal control problem is first solved analytically based on PMP and then numerically with stochastic dynamic programming techniques. In (Malikopoulos et al.,

2018a), the authors address the problem of optimally controlling CAVs crossing an urban intersection without any explicit traffic signaling so as to minimize energy consumption, where a complete analytical solution of the problem is described.

To design or evaluate eco-driving algorithms, an appropriate fuel consumption model that can predict instantaneous (second-by-second) fuel consumption is needed. Faris, in (Faris et al., 2011), performed a comprehensive review of fuel consumption and emission models, such as the VT-Micro model (Ahn et al., 2002), the power-based fuel consumption model (Post et al., 1984), and the POLY model (Teng et al., 2002). On the other hand, instead of using complex fuel consumption models, a strongly simplified cost criterion, namely the square-of-acceleration, is often used by researchers (e.g. (Hong et al., 2017)) to reflect eco-driving aspects, something that is sometimes strongly questioned by other researchers with regard to its relevance to fuel consumption. At first view, the use of this simplified cost criterion seems reasonable due to the positive correlation between the acceleration and the fuel consumption; however, the square-of-acceleration applies equally to decelerations (negative accelerations), which seems less plausible, because fuel consumption is usually low in the deceleration state. Since a comparison between the minimum-fuel vehicle trajectories derived from this simple criterion versus those derived from a genuine fuel consumption model is not available, to the best of our knowledge, the related level of approximation is unknown.

1.2.2 Path-Planning for Automated vehicles

With the recent advances in vehicle communications, either vehicle-to-vehicle (V2V) or infrastructure-to-vehicle (I2V), new communication channels, carrying a potentially wide range of information becomes available in real time. Connected automated vehicles (CAVs) may receive or exchange relevant information, including vehicle state, the current traffic conditions, the next switching time of a traffic signals etc. This extended information entails better knowledge of the driving conditions for CAVs and may be beneficial in various driving situations concerning road safety, flow efficiency and environmental sustainability (Sjoberg et al., 2017; Tian et al., 2018). Specifically, automated driving may improve various driving aspects, such as lane-changing, obstacle avoidance, forming of platoons with short inter-vehicle distances and the application of smoother acceleration or deceleration.

The development of fully automated driving algorithms is inherently related to planning and updating a vehicle path, which should be efficient, collision-free and user-acceptable. Planning such a path can be viewed as a trajectory generation problem, i.e., creation in real time of a quasi-continuous sequence of states that must be tracked by the vehicle via appropriate steering, throttle and braking actions. Inspired by earlier studies on motion planning of robot vehicles in other contexts (e.g. (González et al., 2015)) and driven by imminent implementations and optimistic forecasts of CAV technologies (e.g. (Gu, Hu, 2002;

Nagy, Kelly, 2001)), studies on CAV trajectory optimization in the road traffic context have boomed in the past decade. However, path-planning for road vehicles is a difficult task, since speeds are very high, and safety of the passengers must be guaranteed. Thus, automation on roads calls for sophisticated approaches, such as optimal control methods, advanced feedback control or reinforcement learning (Claussmann et al., 2019; Haydari, Yilmaz, 2020; Karafyllis et al., 2021; Troullinos et al., 2021; Yanumula et al., 2021).

In the related literature, many works exist for trajectory generation of longitudinal motion for CAVs (Wang et al., 2019). Cooperative adaptive cruise control (CACC) or platoon-based systems aim at forming platoons or strings of CAVs, which maintain short inter-vehicle distances. By this formation, vehicles are able to achieve reduced energy/fuel consumption due to the mitigation of aerodynamic drag, as well as increased capacity of the highways (Dey et al., 2015; Wang et al., 2018b). Several methodologies have been proposed for the CACC systems, including linear feedback control (Bernardo et al., 2015; Van Arem et al., 2006; Wang et al., 2017a) and optimal control (Turri et al., 2016; Van De Hoef et al., 2017; Wang et al., 2017b). Cooperative Merging at Highway On-Ramps focuses on addressing the difficult task of how the vehicles, driving along an on-ramp, should adapt their speed in order to enter the main lane in a safe and smooth way. On the other hand, vehicles on the main highway also contribute to this process by modifying their speeds, if necessary, so as to facilitate the merging vehicles, which may affect the traffic flow (Milanés et al., 2010). In the literature, there are numerous studies proposing different control algorithms for the cooperative merging of CAVs (Ntousakis et al., 2016; Rios-Torres, Malikopoulos, 2016a,b; Zhao et al., 2019). Speed harmonization techniques on highways aim at reducing congestion by regulating traffic speeds upstream the congestion point. This is achieved with the application of various highway traffic management strategies, which aim at lowering speeds of traffic upstream the congested area in order to avoid abrupt acceleration/deceleration, which may lead to traffic waves, decrease of safety and traffic flow (Ghiasi et al., 2017; Ma et al., 2016; Malikopoulos et al., 2018b).

On the other hand, there are relatively few studies on CAV trajectory optimization considering lane-changing (Wang et al., 2018c), which indicates that optimization of CAV trajectories in both longitudinal and lateral directions is worth additional investigations. Existing literature indicates that the majority of trajectory planning methods, considering overtaking of moving or static obstacles, employ cell decomposition, potential fields, and optimal control techniques (Dixit et al., 2018).

Cell decomposition, like Rapidly-exploring Random Tree (RRT), is a widely used obstacle avoidance method in robotics, which generates paths within a set of finite, non-overlapping, convex polygons (cells). The set of cells is obtained by decomposing the obstacle-free space (Kuwata et al., 2008; Ma et al., 2014). These algorithms have been modified and applied for CAVs featuring collision-free paths. On the other hand, the memory and computational

requirements are high as the complexity of the algorithms increases, in correlation with the increase of the traffic density (Glaser et al., 2010; Ma et al., 2014). Note also that, the obtained trajectories from RRT's are difficult to be tracked by tracking methods as they are jerky, jeopardizing passengers comfort (Katrakazas et al., 2015).

Potential field methods exploit the idea of repulsive and attractive fields for the corresponding obstacles and collision-free areas, respectively. The vehicle's path is calculated by following the steepest gradient in the generated field (Glaser et al., 2010; Kitazawa, Kaneko, 2016). However, in this methods, the accurate knowledge of the position of static or moving obstacles is important as it may affect the accuracy of the resulting potential fields, and subsequently, affect the vehicles' trajectories.

Optimal control methods aim is to calculate the optimal path of a vehicle by minimizing a performance index, subject to a set of constraints. The performance index may account on different goals, such as minimization of fuel consumption; longitudinal and lateral jerk or/and acceleration; lateral acceleration, etc., while the constraints express, vehicles' speed and acceleration bounds, environment constraints (e.g. road boundaries) or surrounding vehicles (obstacles). Literature results demonstrate that, optimal control methods feature low computational times, while generating collision-free trajectories (Chu et al., 2012; Makantasis, Papageorgiou, 2018; Shim et al., 2012; Werling et al., 2010). On the other hand, the open-loop solution results on trajectories that do not account on the uncertainties that may arise due to the dynamic environment.

Model Predictive Control (MPC) method is, also, widely used for CAVs trajectory planning. MPC has a long history of applications in control, automation, and chemical industries (Mayne, 2014; Mayne, Michalska, 1988; Qin, Badgwell, 2003). The approach solves a finite horizon open-loop optimal control problem to determine a sequence of control inputs; the first part of the optimal inputs is then applied, using a receding horizon principle, and the procedure is repeated (Carvalho et al., 2013). Due to the strongly dynamic environment of the path-planning problem for AVs, the accurate prediction of the surrounding vehicles' paths is crucial, particularly at high highway speeds. However, the uncertainties related to the changing environment and to the actual vehicle advancement inevitably increase over time. Therefore, the MPC approach is often used for online path planning of AVs, as it is very efficient in handling the arising uncertainties due to its re-planning ability (Clausmann et al., 2019; Dixit et al., 2019; Jalalmaab et al., 2015; Rasekhipour et al., 2016; Wang et al., 2018a). One possibility for the MPC procedure is to solve the optimal control problem at each time step, with a corresponding shift of the planning horizon (Howard et al., 2010; Murillo et al., 2018; Nilsson et al., 2015). An alternative possibility is to update path decisions in event-based mode (Earl, D'Andrea, 2007; Khazaeni, Cassandras, 2016).

As far as the connectivity capabilities of the CAVs is concerned, most of the related mentioned works consider limited knowledge of their surroundings, specifically knowledge of

the current position and speed of surrounding vehicles, e.g. through their own sensors. Thus, the short-term future path of the obstacles is considered as a projection of their initial states, e.g. assuming zero acceleration and no lane change. For example, in (Sadat et al., 2020; Shao et al., 2021), the prediction of the surrounding environment is handled through marginal distributions of semantic occupancy over time; or considered as static region, respectively. In the same context, (Saxena et al., 2020) studies the problem of successfully executing a safe and comfortable merge into dense traffic, while the surrounding obstacle vehicles are handled either as static or as constant-speed obstacles. Note that a variety of path planning approaches that emerged in decade-long research in the Robotics domain are valuable for road vehicle path planning as well, but there is a crucial difference in the respective requirements due to automated highway vehicles (and dynamic obstacles) driving much faster than typically considered autonomous robots. This important difference calls for enhanced prediction of the obstacles behaviour, which is indeed achieved via vehicle connectivity, as considered in this work.

1.2.3 Green Light Optimal Speed Advisory (GLOSA) Systems

Consider a vehicle approaching a red traffic light at a given speed. If the vehicle continues driving at that speed, it may reach the traffic light after it has switched to green, in which case no fuel-intensive acceleration needs to be applied; however, the constant-speed vehicle might also reach the traffic light before the green switch, in which case it will have to stop and accelerate to a higher speed after the green switch. On the other hand, if the vehicle decelerates smoothly in view of the red light, this may prove beneficial or not, depending on the time of the green switch. This dilemma of vehicle movement towards a red traffic signal may be addressed by appropriately designed systems. Some early such systems (Van Leersum, 1985) were displaying on road-side dynamic advisory speed signs the speed that would allow a vehicle to cross the downstream signalized junction at green. With recent and emerging advances in vehicle communications, the current state and timing of a traffic signal can be transmitted to equipped vehicles (or apps therein) to enable sensible approaching speed decisions. Based on signal information, it is possible to guide the driver (or an automated vehicle) all the way from the current state to the traffic signal by giving speed advice, which ensures that, by the time the vehicle reaches the traffic signal, it will be green; or, more specifically, that the fuel consumption and emissions will be minimized. Systems (or apps) optimizing the vehicle approach to traffic lights are often referred to as Green Light Optimal Speed Advisory (GLOSA) systems (Stahlmann et al., 2016).

In the case of fixed signals, and hence prior knowledge of the next switching time, a corresponding message is broadcasted by the signal controller. Under these conditions, the problem of how to optimize the approach to traffic signals has been addressed in different

ways. In (Richter, 2005), speed profiles have been compared to their energy demand; however, only a limited set of profiles has been considered, and, more generally, there is a lack of mathematical justification that the derived speed profile is indeed optimal. Rule-based algorithms have been employed in various works to produce advisory speeds for vehicles approaching traffic signals, so as to reduce fuel consumption and emissions, see e.g. (Katsaros et al., 2011; Ma et al., 2018; Sanchez et al., 2006); clearly, rule-based algorithms may deliver sub-optimal results, in particular when the dynamic vehicle kinematics are not accounted for, i.e. when no vehicle acceleration, but only the vehicle speed, is involved. Optimal control approaches, considering explicitly the vehicle kinematics, are, by their nature, more pertinent in producing fuel-optimal speed profiles for vehicles approaching fixed traffic signals. Such an approach is proposed in (Lawitzky et al., 2013), attempting the minimization of vehicle's physical work; the resulting optimal control problem is solved analytically via Pontryagin's Minimum Principle (PMP). This paper also presents an optimal control approach for fuel minimization with more pertinent final state conditions for the vehicle, consisting of a final position sufficiently downstream of the signals and a high final speed, so as to avoid distorted or non-comparable vehicle trajectories.

The situation becomes more complicated when real-time signals with very short (e.g. second-by-second) control update periods are present, in which case exact prior knowledge of the next switching time is not available even with the signal controller. In this case, the best available knowledge about the next signal switching can be presented as an estimate or as a probabilistic distribution for the next switching time within a short-term future time-window. A system called SignalGure (Koukoumidis et al., 2011) does not require communication to the infrastructure, but relies solely on a collection of mobile phone data to detect and predict the traffic signal timing. The mobile phones detect current traffic signals with their cameras, collaboratively communicate and learn traffic signal timing patterns, and predict their future timing. Reportedly (Koukoumidis et al., 2011), SignalGuru's estimates come on average within 0.66 s for fixed traffic signals; and within 2.45 s for real-time traffic signals; of the actual switching times. Such an estimate of the next switching time may then be used as a proxy for any of the GLOSA approaches that need a known switching time, possibly with appropriate extensions to account for the consequences of estimation inaccuracy. Richer probabilistic information than a single estimate for the next signal switching time can be produced in form of a probability distribution for the next switching time within a short-term future time-window by use of statistics from previous signal operations over a sufficiently long rolling horizon. Such an approach to produce probability distribution for the next switching time, within a short-term future time-window, based on past signal switching, is proposed in (Mahler, Vahidi, 2012). However, the available probability distribution is used heuristically (rather than optimally), to accordingly time-weigh the objective function, within a deterministic optimal control problem that is solved via a Dynamic Programming algorithm.

In (Sun et al., 2020), the uncertainty of traffic signal timing is considered as a data-driven chance-constrained robust optimization problem, where the red-light duration is defined as a random variable. A data-driven approach is adopted to formulate chance constraints, based on empirical sample data, and Dynamic Programming is employed to solve the optimization problem. An appropriate utilization of the available probabilistic distribution of the next signal switching time within a stochastic optimal control problem is proposed in (Lawitzky et al., 2013), and the problem is solved via a discrete Stochastic Dynamic Programming (SDP) algorithm; however, the formulated optimal control problem extends only up the point in time where the next switching time becomes known, thus neglecting the cost incurred after this time until the vehicle's final state is reached, something that may lead to non-commensurable vehicle trajectories and skewed solutions.

Chapter 2

Minimization of Fuel Consumption for Vehicle Trajectories

2.1 Abstract

The importance of eco-driving in reducing cumulative fuel consumption of road vehicles is a well-known and widely treated subject. Eco-driving intends to minimize fuel consumption by maneuvering a vehicle with a human or automated driver. In this work, the eco-driving problem is cast in an optimal control framework with fixed time horizon. State equations reflect the simple vehicle kinematics for position and speed, with the acceleration acting as a control input. Initial and final states (position and speed) are fixed. For the fuel consumption estimation, a number of alternatives are employed. To start with, a realistic, but nonlinear and non-smooth formula from the literature is considered. Simple smoothing procedures are then applied, so as to enable the application of powerful numerical algorithms for the efficient solution of the resulting nonlinear optimal control problem. Furthermore, suitable quadratic approximations of the nonlinear formula are also considered, which enable analytical problem solutions. A comprehensive comparison on the basis of various driving scenarios demonstrates the features of each alternative approach. In particular, it is demonstrated that the often utilized, but sometimes strongly questioned, square-of-acceleration term delivers excellent approximations for fuel-minimizing trajectories in the present setting.

2.2 Introduction

Energy and fuel consumption reduction has gained high importance due to the related reduction of emissions, among other reasons. The current trend towards increased mobility seems to run counter to the purposes of controlling the greenhouse effect, local pollution and the efficient exploitation of fuel resources; as a result, fuel consumption has become a crucial

issue. Intensive research has been carried out in order to find optimal solutions for a number of possible settings.

Research related to vehicle control for improving its fuel economy has a long track. Some early studies were conducted to determine the optimal cruising velocity of a vehicle based on its operating characteristics (Gilbert, 1976). In other studies, such as (Hooker, 1988), the optimal values of acceleration for starting up and cruising speed over hilly terrains or flat roads are determined for fuel-efficient driving. The problem of improving fuel economy of road vehicles has been recently often referred to as eco-driving (Barkenbus, 2010). Ecological (eco) driving is a way of maneuvering a vehicle with a human (or, more recently, an automated) driver that is intended to minimize fuel consumption by trading off the most efficient driving behaviour of the vehicle. Aside from various physical factors, driving style has a great influence on vehicle emissions and energy consumption (Van Mierlo et al., 2004). Several studies have indicated that eco-driving can improve fuel economy by 15-25% (Kamal et al., 2011).

The eco-driving problem can be formulated as an optimal control problem. To solve the problem, one may employ Dynamic Programming (DP) techniques (Hooker, 1988; Mensing et al., 2011; Monastyrsky, Golownykh, 1993; Ozatay et al., 2014); however, DP application may face real-time feasibility difficulties, in cases where online solutions are required, due to the method's notorious "curse of dimensionality". Alternatively, one may transform the optimal control problem (in a number of possible ways) to a Nonlinear Programming (NLP) problem that can be solved using numerical NLP methods, see e.g. (Saboohi, Farzaneh, 2009; Xu et al., 2014); note, however, that even some NLP-based approaches may lead to computational times that are not feasible in real-time applications.

Another approach for the solution of the fuel/energy efficiency problem is to use Pontryagin's Minimum Principle (PMP), where the optimal control problem is transformed to a boundary value problem (Petit, Sciarretta, 2011; Schwarzkopf, Leipnik, 1977). In (Ozatay et al., 2012), the authors develop an analytical solution to the optimization problem of finding the minimum fuel consumption of a vehicle given the road and possibly the traffic light information. Also, the authors in (Kim et al., 2010) proposed an optimal control based on PMP approach. The simulation results are the same as the global optimal control based on dynamic programming results. The work in (Lawitzky et al., 2013) concentrates on energy optimal approaching red traffic lights. The energy efficiency optimal control problem is first solved analytically based on PMP and then numerically with stochastic dynamic programming techniques. In (Malikopoulos et al., 2018a), the authors address the problem of optimally controlling CAVs crossing an urban intersection without any explicit traffic signaling so as to minimize energy consumption, where a complete analytical solution of the problem is described.

To design or evaluate eco-driving algorithms, an appropriate fuel consumption model that can predict instantaneous (second-by-second) fuel consumption is needed. Faris, in (Faris et al.,

2011), performed a comprehensive review of fuel consumption and emission models, such as the VT-Micro model (Ahn et al., 2002), the power-based fuel consumption model (Post et al., 1984), and the POLY model (Teng et al., 2002). On the other hand, instead of using complex fuel consumption models, a strongly simplified cost criterion, namely the square-of-acceleration, is often used by researchers (e.g. (Hong et al., 2017)) to reflect eco-driving aspects, something that is sometimes strongly questioned by other researchers with regard to its relevance to fuel consumption. At first view, the use of this simplified cost criterion seems reasonable due to the positive correlation between the acceleration and the fuel consumption; however, the square-of-acceleration applies equally to decelerations (negative accelerations), which seems less plausible, because fuel consumption is usually low in the deceleration state. Since a comparison between the minimum-fuel vehicle trajectories derived from this simple criterion versus those derived from a genuine fuel consumption model is not available, to the best of our knowledge, the related level of approximation is unknown.

This research aims at the minimization of the vehicle fuel consumption by optimizing its kinematic trajectories (using vehicle acceleration as the control input), while driving from a given initial state to a given final state within a fixed time horizon. The minimization problem is formulated as an optimal control problem, which minimizes a specific energy-related, realistic but complex fuel consumption function (Akçelik, Biggs, 1987). In a first (approximate) solution approach, the problem is solved analytically, based on Pontryagin's Minimum Principle, using a quadratic approximation of the fuel consumption function. As a special case of this solution, we also obtain the vehicle trajectories which minimise the square-of-acceleration criterion. On the other hand, in a second (exact) solution approach, we consider the optimal control problem with the original complex nonlinear fuel consumption function, appropriately smoothed at its non-smooth locations, and proceed to the numerical solution of the optimal control problem by use of a very efficient feasible direction algorithm (Papageorgiou et al., 2015, 2016). Specifically, we use the reduced gradient to enable the problem solution in the reduced space of the control variables, and we employ an NLP-like iterative algorithm to produce the solution trajectories. The results of the approximate and exact solutions are compared for a number of driving scenarios, and it is demonstrated that simpler (quadratic) functions may indeed serve as appropriation tools for obtaining good approximations of eco-driving solutions. Note that, part of this work is also included in Typaldos MSc Thesis (Typaldos, 2017), however the present work includes additional and more extensive evaluations and investigations over more realistic scenarios.

The remainder of the paper is organized as follows: Section 2.3 presents the fuel consumption model used as a cost criterion. The fuel consumption optimal control problem is formulated and solved in Section 2.4. Several variations and comparison results of the approximate and exact solutions are presented in Section 2.5, and Section 2.6 concludes this work.

2.3 Fuel Consumption Criterion

2.3.1 Energy-Related Model for Estimating Fuel Consumption

The model used for the estimation of the vehicle's fuel consumption rate is the Australian Road Research Board (ARRB) model (or Energy-Related model) described below, which estimates instantaneous values of fuel consumption in dependence of the current speed, acceleration and grade information. The model is an extended and modified version of the power model described in (Post et al., 1984), with more details given in (Akçelik, Biggs, 1987) and (Bowyer et al., 1985). Basically, the model relates fuel consumption during a small increment, dt , to:

- (a) the fuel to maintain engine operation,
- (b) the energy consumed (work done) by the vehicle engine while traveling an increment of distance dx during this time period, and
- (c) the product of energy and acceleration during periods of positive acceleration.

Part (c) accounts for the inefficient use of fuel during periods of high acceleration. Since energy is $dE = R_T dx$, where R_T is the total tractive force required to drive the vehicle along distance dx , the fuel consumed in the time increment dt is expressed as

$$dF = \begin{cases} \alpha dt + \beta_1 R_T dx + [\beta_2 a R_T dx]_{a>0} & \text{for } R_T > 0 \\ \alpha dt & \text{for } R_T \leq 0 \end{cases} \quad (2.1)$$

where

dF is the increment of fuel consumed (ml) during travel along distance dx (m) and in time dt (s),

α is the constant idle fuel rate (ml/s), which applies during all modes of driving as an estimate of fuel needed to maintain engine operation,

β_1 is an efficiency parameter which relates fuel consumed to the energy provided by the engine, i.e. fuel consumption per unit of energy (ml/kJ),

β_2 is an efficiency parameter, which relates fuel consumed during positive acceleration to the product of inertia energy and acceleration, i.e. fuel consumption per unit of energy-acceleration (ml/(kJ · m/s²)),

a is the instantaneous acceleration (dv/dt) in m/s^2 , which has a negative value for slowing down, and

R_T is the total "tractive" force required to drive the vehicle, which is the sum of drag force (R_D), inertia force (R_I) and grade force (R_G) in kN (kilonewtons):

$$R_T = R_D + R_I + R_G. \quad (2.2)$$

The resistive forces can be expressed as:

$$R_D = b_1 + b_2v^2 \quad (2.3)$$

$$R_I = Ma/1000 \quad (2.4)$$

$$R_G = 9.81M(G/100)/1000 \quad (2.5)$$

where

v is the speed (dx/dt) in m/s ,

G is the percent grade which has a negative value for downhill grade,

M is the vehicle mass in kg, including occupants and any other load, and

b_1, b_2 are the vehicle parameters related mainly to rolling resistance and aerodynamic drag, but each containing a component due to drag associated with the engine. The parameters of drag function (2.3) were derived using steady-speed fuel consumption data. However, if data collected during coast-down in neutral are also available, a three-term function $R_D = b_1 + b_2v + b_3v^2$ can be derived where b_1, b_2 and b_3 are related to rolling, engine and aerodynamic drag, respectively.

Instantaneous fuel consumption per unit time (ml/s) can then be expressed as:

$$f(v, a) = \frac{dF}{dt} = \begin{cases} \alpha + \beta_1 R_T v + \left[\frac{\beta_2 M a^2 v}{1000} \right]_{a>0} & \text{for } R_T > 0 \\ \alpha & \text{for } R_T \leq 0 \end{cases} \quad (2.6)$$

where the total tractive force required is:

$$R_T = b_1 + b_2v + b_3v^2 + \frac{Ma}{1000} + 9.81 \times 10^{-5}MG. \quad (2.7)$$

The following parameter values derived for the Melbourne University test car (4.1-L Ford Cortina station-wagon) with automatic transmission (Biggs, Akçelik, 1986) have been used for

this work: $M = 1600$ kg, $G = 0$, $\alpha = 0.666$ ml/s, $\beta_1 = 0.0717$ ml/kJ, $\beta_2 = 0.0344$ ml/(kJ·m/s²), $b_1 = 0.269$ kN, $b_2 = 0.0171$ kN(m/s)⁻², $b_3 = 0.000672$.

2.3.2 Smooth Form of the ARRB Fuel Consumption Function

The ARRB fuel consumption function (2.6) can be expressed in the form $f(v, a) = \max\{A_1, A_2\}$, assuming $v \geq 0$, where

$$\begin{aligned} A_1 &= \alpha \\ A_2 &= \begin{cases} \alpha + \beta_1 R_T v + \frac{\beta_2 M a^2 v}{1000} & \text{for } a > 0 \\ \alpha + \beta_1 R_T v & \text{for } a \leq 0 \end{cases} \end{aligned} \quad (2.8)$$

Consider a general non-smooth function $F = \max\{A_1, \dots, A_n\}$. As suggested in (Jamshidnejad et al., 2015), such a function may be approximated for $\alpha_s \gg 1$, via a smooth function as follows:

$$F = \max_{i=1, \dots, n} \{A_i\} \approx \frac{1}{\alpha_s} \log \sum_{i=1}^n e^{\alpha_s A_i} \quad (2.9)$$

with α_s a smoothing parameter. The bigger one chooses the parameter α_s , the closer is the smooth approximation to the non-smooth max-function.

By applying (2.9) and (2.8) to the ARRB fuel consumption model we obtain the smooth approximation

$$f_s(v, a) \approx \begin{cases} \frac{1}{\alpha_s} (\log(e^{\alpha_s \alpha} + e^{\alpha_s (\alpha + \beta_1 R_T v + \frac{\beta_2 M a^2 v}{1000})})) & \text{for } a > 0 \\ \frac{1}{\alpha_s} (\log(e^{\alpha_s \alpha} + e^{\alpha_s (\alpha + \beta_1 R_T v)})) & \text{for } a \leq 0 \end{cases} \quad (2.10)$$

Note that the original ARRB function is actually smooth at $a = 0$, hence the smoothing approximation is only applied at $R_T = 0$. For the choice of the parameter α_s , different values have been tested to observe the behavior of the smoothed model. As expected, and can be verified in Fig. 2.1, the smaller the value of the parameter α_s is (see the case where $\alpha_s = 1$), the more the smoothed model diverges from the original model, while as the value is increased, the smoothed-function approaches increasingly the non-smooth fuel consumption model. The performance of the approximation with different smoothing parameters is similar for different speed values. The value $\alpha_s = 20$ was finally selected, since the fuel consumption differences, when using higher values, were practically insignificant.

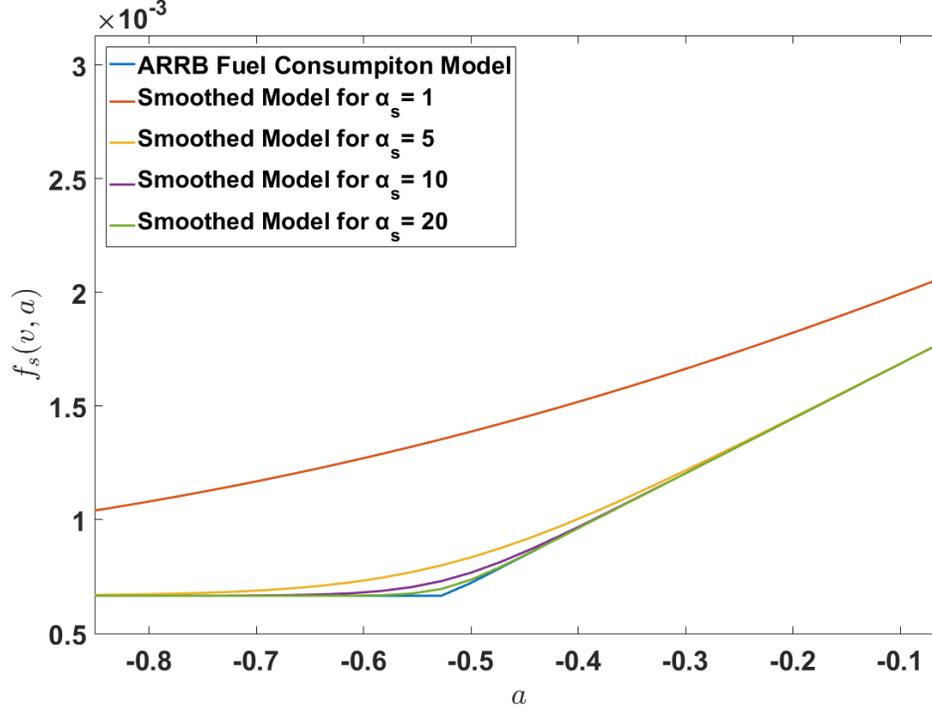


Figure 2.1: The ARRB Fuel Consumption function and its smoothed variants versus the acceleration a (assuming the value of speed is constant, $v = 15$ m/s) for different values of α_s .

2.3.3 Quadratic Approximation of the Fuel Consumption Model

A quadratic approximation can be achieved through a Taylor expansion, approximating the fuel consumption function around a given point ($v = v_k, a = a_k$). The quadratic form of a second-order Taylor polynomial can be expressed in general form as follows

$$\begin{aligned}
 q(v, a) = & f(v_k, a_k) + \nabla f(v_k, a_k)^T \begin{bmatrix} v - v_k \\ a - a_k \end{bmatrix} \\
 & + \frac{1}{2} [v - v_k \quad a - a_k]^T \nabla^2 f(v_k, a_k) \begin{bmatrix} v - v_k \\ a - a_k \end{bmatrix}
 \end{aligned} \tag{2.11}$$

where $\nabla f(v_k, a_k)$ denotes the gradient vector and $\nabla^2 f(v_k, a_k)$ the Hessian matrix of $f(v, a)$ at the point ($v = v_k, a = a_k$).

By defining a general form of the gradient vector and the Hessian matrix of the approximated model

$$\nabla f(v_k, a_k) = [g_1 \ g_2]^T \quad (2.12)$$

$$\nabla^2 f(v_k, a_k) = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \quad (2.13)$$

where g_1, g_2 and h_{11}, \dots, h_{22} are the elements of the gradient vector and the Hessian matrix, respectively, the quadratic Taylor approximation function becomes

$$\begin{aligned} q(v, a) = & \frac{a^2 h_{22}}{2} + \frac{v^2 h_{11}}{2} + \\ & v \left(g_1 + \frac{1}{2} a (h_{12} + h_{21}) - \frac{1}{2} a_k (h_{12} + h_{21}) - h_{11} v_k \right) \\ & + a \left(g_2 - a_k h_{22} - \frac{1}{2} (h_{12} + h_{21}) v_k \right) \\ & + \frac{1}{2} (2f(v_k, a_k) - 2a_k g_2 + a_k^2 h_{22} \\ & + a_k (h_{12} + h_{21}) v_k + v_k (-2g_1 + h_{11} v_k)) \end{aligned} \quad (2.14)$$

This function may be used as an approximation of the nonlinear model $f(v, a)$ within the optimal control problem.

2.4 Optimal Control Problems

The problem considered in this work is a typical case of a fuel consumption minimization problem, where a vehicle starts from an initial state $\mathbf{x}_0 = [x_0, v_0]^T$, where x_0 is a given initial position and v_0 a given initial speed of the vehicle, and must reach a final state $\mathbf{x}_e = [x_e, v_e]^T$ within a fixed time horizon T , with x_e and v_e being the fixed final position and speed of the vehicle, respectively. The objective of the vehicle, is to appropriately adjust its acceleration in order to achieve the minimum possible fuel consumption, while satisfying the initial and final conditions \mathbf{x}_0 and \mathbf{x}_e traveling alone on a straight road segment.

The minimization problem outlined above is formulated as an optimal control problem. The system is described by the following state variables:

$$\dot{x} = v \text{ (speed)} \quad (2.15)$$

$$\dot{v} = a \text{ (acceleration)}. \quad (2.16)$$

The objective is to drive the system from the given initial condition $\mathbf{x}_0 = [x_0, v_0]^T$ to the fixed final condition $\mathbf{x}_e = [x_e, v_e]^T$ within the given time T , while minimizing the criterion

$$J = \int_0^T f(v, a) dt. \quad (2.17)$$

If necessary, upper and lower bounds may be applied to speed v and acceleration a . Such constraints would limit the speed between zero and its maximum value; and the acceleration between its minimum and maximum values.

2.4.1 Approach 1: Analytic Solution of Approximated Fuel Consumption Model

As a first approach, a simplified version of the optimal control problem is solved analytically. To enable the analytic solution, one may approximate the fuel consumption function with a quadratic function (see 2.3.3).

2.4.1.1 Optimal Control Methodology for the Approximated Fuel Consumption Model

The approximate fuel consumption optimal control problem considers the minimization of the approximated quadratic fuel consumption (2.14), replacing $f(v, a)$ in (2.17) subject to the state equations of vehicle motion, (2.15) and (2.16), with fixed initial and final states $x(0) = x_0$, $x(T) = x_e$, $v(0) = v_0$, $v(T) = v_e$.

The quadratic cost criterion leads to smooth variations of the states and the control over time. For the sake of simplicity in deriving the optimal solution, no constraints are considered for the state and control variables. It should be noted that the derivation of analytic solutions for the corresponding constrained optimization problem is not an easy task and may indeed be even infeasible or computationally demanding due to many possible combinations of subsequent constraint activation and deactivation that would need to be elaborated. In any case, a non-analytic computation possibility, with explicit constraint consideration, is offered in the next sections.

The Hamiltonian function (Pontryagin, 1987) for the approximate optimal control problem reads

$$H(v, a, \lambda_1, \lambda_2) = q(v, a) + \lambda_1 v + \lambda_2 a \quad (2.18)$$

where λ_1 and λ_2 are the co-state variables.

The optimal control is calculated from

$$\begin{aligned} \frac{\partial H}{\partial a} = 0 \Rightarrow \\ g_2 + \frac{1}{2}(2(a - a_k)h_{22} + h_{12}(v - v_k) + h_{21}(v - v_k)) + \lambda_2 = 0. \end{aligned} \quad (2.19)$$

Solving (2.19) for a , yields

$$a = -\frac{(h_{12} + h_{21})v}{2h_{22}} + \frac{-2g_2 + 2a_k h_{22} + (h_{12} + h_{21})v_k}{2h_{22}} - \frac{\lambda_2}{h_{22}}. \quad (2.20)$$

Substituting (2.20) into (2.18), the optimal Hamiltonian function $H_0 = H(v, \lambda_1, \lambda_2)$ is derived.

The other necessary conditions of optimality are:

$$\dot{x} = \frac{\partial H_0}{\partial \lambda_1} = v \quad (2.21)$$

$$\begin{aligned} \dot{v} = \frac{\partial H_0}{\partial \lambda_2} = & -\frac{(h_{12} + h_{21})v}{2h_{22}} \\ & + \frac{-2g_2 + 2a_k h_{22} + (h_{12} + h_{21})v_k}{2h_{22}} - \frac{\lambda_2}{h_{22}} \end{aligned} \quad (2.22)$$

$$\dot{\lambda}_1 = -\frac{\partial H_0}{\partial x} = 0 \quad (2.23)$$

$$\begin{aligned} \dot{\lambda}_2 = -\frac{\partial H_0}{\partial v} = & \left(-h_{11} + \frac{(h_{12} + h_{21})^2}{4h_{22}} \right) v \\ & - \lambda_1 + \frac{(h_{12} + h_{21})\lambda_2}{2h_{22}} \\ & - \frac{-2g_2(h_{12} + h_{21}) + 4g_1 h_{22} + \left((h_{12} + h_{21})^2 - 4h_{11}h_{22} \right) v_k}{4h_{22}} \end{aligned} \quad (2.24)$$

The system of ordinary differential equations (2.21)-(2.24) is linear and can be easily solved analytically. Its solution is given by an exponential-function of time for the involved state and co-state variables. The related parameters of the solution trajectories depend on the initial and final conditions of the problem as well as on the approximation point used for the fuel consumption model. This solution can be easily derived in analytic (explicit) form using symbolic differentiation tools (e.g. Mathematica (Wolfram Research, Inc., 2016)). The analytic solution for $x(t)$, $v(t)$, $\lambda_1(t)$, $\lambda_2(t)$ may then be plugged into (2.20) to get the analytic

solution for the optimal acceleration $a(t)$.

2.4.1.2 Approach 1-a: A Special Case of the Approximated Fuel Consumption Model

It is important to note that, a special case, that is often used (e.g. (Hong et al., 2017)) as a proxy for fuel consumption minimization, can be derived from the general quadratic solution (2.14). This special case is the simple quadratic cost criterion

$$\tilde{q}(v, a) = \frac{1}{2}a^2 \quad (2.25)$$

with the corresponding gradient vector and the Hessian matrix ((2.12) and (2.13)) of the approximated fuel consumption model equal to

$$\nabla f(v, a) = [0 \quad a]^T \quad \text{and} \quad \nabla^2 f(v, a) = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.26)$$

This cost function is often used as a surrogate of fuel consumption, on the grounds that there is a positive correlation between acceleration and fuel consumption. However, as the original formula in (2.6) indicates, this statement does not apply in case of vehicle deceleration (i.e. negative acceleration); hence it is of significant interest to test and compare the produced optimal results with those produced by use of the original complex fuel consumption model, so as to conclude on the appropriateness (or otherwise) of, and level of approximation resulting from the minimization of the square-of-acceleration for eco-driving problems.

In this special case, the solution trajectory for the control (acceleration) is linear with respect to time t (see (Ntousakis et al., 2016)). In fact, we have from (2.23) $\lambda_1(t) = c_1$, where c_1 is a constant; while, for this special case, (2.24) yields $\dot{\lambda}_2 = -\lambda_1$, hence $\lambda_2(t) = -c_1t + c_2$, where c_2 is another constant. Furthermore, (2.20) simplifies in this special case to $a = -\lambda_2$, hence we have finally

$$a(t) = c_1t - c_2 \quad (2.27)$$

i.e. a linear function of time, where c_1 and c_2 may be calculated from the boundary conditions.

It is pointed out that both Approach 1 and 1-a produce a closed form solution that depends on the initial and final conditions of the problem and can be applied instantaneously.

2.4.2 Approach 2: Numerical Solution of the Smoothed Fuel Consumption Model

The analytic solution of optimal control problems by use of the necessary conditions of optimality is only possible for simple problems, e.g. problems with low dimension or

problems with special structure (e.g. Linear-Quadratic Optimization). In all other cases, numerical solution algorithms are required for the calculation of the optimal trajectories. For the purposes of this work, a very efficient feasible direction algorithm has been used (Papageorgiou et al., 2016), which exploits the notion of reduced gradient to eliminate the state variables and solve the problem in the space of the control variables using well-known NLP methods.

It is noted that the ARRB fuel consumption function (2.6) is a continuous, but non-smooth function, since it features discontinuous first-order derivatives when the value of the total force term R_T is equal to 0. This may lead to difficulties when using gradient-based optimization methods. Hence, it would be useful to have a smooth nonlinear cost function that is differentiable everywhere in its domain, as the one derived in Section 2.3.2.

2.4.2.1 Feasible Direction Algorithm

The feasible direction algorithm can be applied to continuous-time or discrete-time formulations of an optimal control problem. For the convenience of using available software, we employ the discrete-time version of the algorithm. Then, the general optimal control problem that is addressed by the method is as follows.

We consider a discrete-time dynamic process described by the following set of state-space difference equations organized in vector form

$$\mathbf{x}(k+1) = \mathbf{f}[\mathbf{x}(k), \mathbf{u}(k)], \quad k = 0, \dots, K-1 \quad (2.28)$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$ are the system state and control variable vectors, respectively, k is the discrete time index, K is the considered discrete time horizon, and $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is a twice continuous differentiable cost function. The system has a known initial state

$$\mathbf{x}(0) = \mathbf{x}_0. \quad (2.29)$$

The problem consists in minimizing the discrete-time cost function

$$J = \boldsymbol{\theta}[\mathbf{x}(K)] + \sum_{k=0}^{K-1} \phi[\mathbf{x}(k), \mathbf{u}(k)] \quad (2.30)$$

subject to (2.28) and (2.29), where $\boldsymbol{\theta} : \mathbb{R}^n \rightarrow \mathbb{R}^l$, $\phi : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^q$ are twice continuous differentiable vector functions. The final state must be free. Upper and lower bounds may apply to the control variables $\mathbf{u}(k)$.

The solution of a discrete-time optimal control problem corresponds to the calculation of the optimal trajectories for the control variables, $\mathbf{u}(k)$, and corresponding states, $\mathbf{x}(k+1)$,

$k = 0, \dots, K - 1$, that minimize the cost function over the time horizon. To describe the solution procedure, we need to first present the necessary optimality conditions for the formulated discrete-time optimal control problem. As it is customary, we use to this end the discrete-time Hamiltonian Function

$$H[\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\lambda}(k+1)] = \phi[\mathbf{x}(k), \mathbf{u}(k)] + \boldsymbol{\lambda}(k+1)^T \mathbf{f}[\mathbf{x}(k), \mathbf{u}(k)] \quad (2.31)$$

where $\boldsymbol{\lambda}(k+1)$ stands for the co-state vector.

Then we have the following conditions of optimality that must be satisfied for $k = 0, \dots, K - 1$ (notation: $x_y = dx/dy$):

$$\mathbf{x}(k+1) = H_{\boldsymbol{\lambda}(k+1)} = \mathbf{f}[\mathbf{x}(k), \mathbf{u}(k)] \quad (2.32)$$

$$\boldsymbol{\lambda}(k) = H_{\mathbf{x}(k)} = \phi_{\mathbf{x}(k)}(k) + \mathbf{f}_{\mathbf{x}(k)}^T \boldsymbol{\lambda}(k+1) \quad (2.33)$$

$$H_{\mathbf{u}(k)} = \phi_{\mathbf{u}(k)}(k) + \mathbf{f}_{\mathbf{u}(k)}^T \boldsymbol{\lambda}(k+1) = 0 \quad (2.34)$$

Furthermore, the following boundary conditions must be satisfied

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (2.35)$$

$$\boldsymbol{\lambda}(K) = \boldsymbol{\theta}_{\mathbf{x}(K)} \quad (2.36)$$

(for more details regarding the conditions of optimality, see (Papageorgiou et al., 2015, 2016)).

The solution of this problem is calculated by use of the very efficient feasible direction algorithm (Papageorgiou et al., 2016), which exploits the structure of the state equations and maps the optimal control problem into an NLP problem in the reduced space of control variables. Thus, the algorithm attempts the calculation of a control trajectory $\mathbf{u}(k)$, $k = 0, \dots, K - 1$, which corresponds to a local minimum of the cost function in the mK -dimensional space, where m is the number of control variables. This marks a substantial reduction of the problem dimension, as the state variables are eliminated.

The feasible direction algorithm is an iterative procedure, whose algorithmic steps are presented below (superscripts (i) indicate the iteration index);

Step 1 Guess an initial control trajectory $\mathbf{u}^{(0)}(k)$, $k = 0, \dots, K - 1$; Calculate states $\mathbf{x}^{(0)}(k)$ from (2.32), (2.35) and co-states $\boldsymbol{\lambda}^{(0)}(k)$, $k = 0, \dots, K - 1$ from (2.33), (2.36). Calculate the reduced gradient $\mathbf{g}^{(0)}(k) = \frac{dH}{d\mathbf{u}(k)}$, $k = 0, \dots, K - 1$, using (2.34). Set iteration index $i = 0$.

Step 2 Specify a descent direction $\mathbf{s}^{(i)}(k)$, $k = 0, \dots, K - 1$.

Step 3 Specify an optimal scalar step $\xi^{(i)} > 0$ through line optimization.

Set $\mathbf{u}^{(i+1)}(k) = \mathbf{u}^{(i)}(k) + \xi^{(i)} \mathbf{s}^{(i)}(k)$, $k = 0, \dots, K - 1$.

Calculate states $\mathbf{x}^{(i+1)}(k)$, co-states $\boldsymbol{\lambda}^{(i+1)}(k)$ and gradient $\mathbf{g}^{(i+1)}(k)$, $k = 0, \dots, K - 1$ as above.

Step 4 If a convergence test is satisfied, stop, otherwise start a new iteration $i := i + 1$, i.e. go to Step 2.

The algorithm starts with an initial guess control trajectory $\mathbf{u}^{(0)}(k)$, $k = 0, \dots, K - 1$. At each iteration i , it calculates the corresponding states, $\mathbf{x}^{(i)}(k)$, (from (2.32), (2.35)) and co-states, $\boldsymbol{\lambda}^{(i)}(k)$, (from (2.33), (2.36)). Then, based on the calculated states and co-states, the algorithm computes the reduced gradient $\mathbf{g}^{(i)}(k) = dH/d\mathbf{u}(k)$ (from (2.34)), which is used to specify a descent direction $\mathbf{s}^{(i)}(k)$. Finally, the control trajectory is updated

$$\mathbf{u}^{(i+1)}(k) = \mathbf{u}^{(i)}(k) + \xi^{(i)} \mathbf{s}^{(i)}(k), \quad (2.37)$$

where $\xi^{(i)}$ is a scalar step specified via line optimization. For the specification of the descent search direction $\mathbf{s}^{(i)}(k)$, five methods have been used and compared in this work: two quasi-Newton methods, DFB and BFGS; two conjugate-gradient methods, Fletcher-Reeves and Polak-Ribiere; and the Steepest-Descent method, see (Papageorgiou et al., 2015, 2016) for details.

At the end of each iteration, if a convergence criterion (gradient close to zero) is satisfied, then all necessary conditions are satisfied, and the algorithm stops; otherwise it starts a new iteration. Note that the method does not allow for strictly fixed final states, but these may be readily considered indirectly by the introduction of appropriate penalty terms in the final term $\theta[\mathbf{x}(K)]$ of the objective function (2.30). The algorithm may be readily extended to consider fixed bounds of the control variables, see (Papageorgiou et al., 2015, 2016) for all details.

2.4.2.2 Discrete-Time Optimal Control Problem Formulation

We will now transform the fuel-optimal control problem into a discrete-time optimal control problem as above. The discrete-time version of the state equations (2.15), (2.16) is

$$\begin{aligned} x(k+1) &= x(k) + v(k)\Delta t + \frac{1}{2}a(k)\Delta t^2 \\ v(k+1) &= v(k) + a(k)\Delta t \end{aligned} \quad (2.38)$$

and have indeed the required general form of (2.28), with Δt being the time step length and $k \cdot \Delta t = t$. For the objective function, we have

$$J[v(k), a(k)] = \frac{1}{2} p_1 [x_e - x(K)]^2 + \frac{1}{2} p_2 [v_e - v(K)]^2 + \sum_{k=0}^{K-1} \left\{ \frac{1}{2} p_3 \min[0, v(k) + \varepsilon]^2 + f_s[v(k), a(k)] \right\} \quad (2.39)$$

where the first two quadratic penalty terms account for the final conditions of the vehicle position and speed, respectively, by penalizing deviations from the respective desired final states; and $f_s[v(k), a(k)]$ is the smoothed nonlinear fuel consumption function (2.10). The penalty weights p_1 and p_2 must be selected sufficiently large to ensure the virtual fulfillment of the final state conditions. Moreover, a penalty term to suppress negative speeds has been added in the objective function, with p_3 being the corresponding weight. This additional penalty term ensures that no negative speed values will appear in the solution. Note that this term is mainly activated in the case of the fuel consumption maximization problem that will be addressed later.

2.4.3 A Stop Point Scenario

A slightly modified scenario of the fuel minimization problem is examined in this section, where a vehicle starting from an initial state $\mathbf{x}_0 = [x_0, v_0]^T$ aims at reaching a final state $\mathbf{x}_e = [x_e, v_e]^T$ within a fixed time horizon T , but now with the restriction that it cannot pass through a given position x_1 , before a fixed time t_r (assuming that x_1 is the position of a traffic light and t_r is the time that the traffic light turns green from red, see Fig. 2.2). The objective of the vehicle, also in this case, is to appropriately adjust its acceleration in order to achieve the minimum possible fuel consumption, while satisfying the initial and final conditions as well as the intermediate (traffic signal) constraint.

2.4.3.1 Continuous-Time Formulation

The optimal control problem is described by the same states and the objective is to drive the system from the given initial condition $\mathbf{x}_0 = [x_0, v_0]^T$ to the fixed final condition $\mathbf{x}_e = [x_e, v_e]^T$ within the given time T , while minimizing the criterion

$$J(t_1) = \frac{1}{2} \int_0^{t_1} a^2 dt + \int_{t_1}^T a^2 dt. \quad (2.40)$$

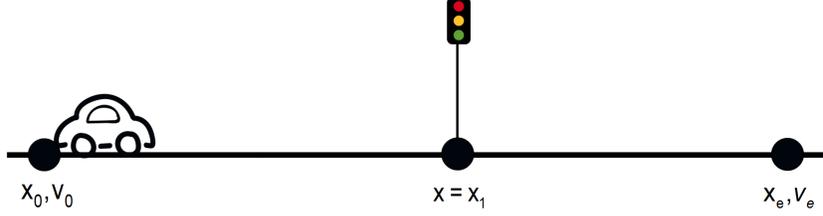


Figure 2.2: A vehicle starting from a given initial state (x_0, v_0) , aims to reach a final state (x_e, v_e) with the presence of a traffic light at a given position that switches to green at time t_r .

i.e., we consider the special case of Approach 1-a for this enlarged problem, in which, in addition, the constraint, $t_r \leq t_1$, must be fulfilled, where t_1 is the time in which the vehicle crosses from the position x_1 .

As for this case, the state and control trajectories can be derived from the solution of the following equation system

$$a(t) = \begin{cases} c_1 t + c_2 & 0 \leq t \leq t_1^- \\ c_5 t' + c_6 & t_1^+ \leq t \leq T \end{cases} \quad (2.41)$$

$$v(t) = \begin{cases} \frac{1}{2} c_1 t^2 + c_2 t + c_3 & 0 \leq t \leq t_1^- \\ \frac{1}{2} c_5 t'^2 + c_6 t' + c_7 & t_1^+ \leq t \leq T \end{cases} \quad (2.42)$$

$$x(t) = \begin{cases} \frac{1}{6} c_1 t^3 + \frac{1}{2} c_2 t^2 + c_3 t + c_4 & 0 \leq t \leq t_1^- \\ \frac{1}{6} c_5 t'^3 + \frac{1}{2} c_6 t'^2 + c_7 t' + c_8 & t_1^+ \leq t \leq T \end{cases} \quad (2.43)$$

where $t' = t - t_1$, using the initial and final conditions of the problem, the corner conditions of continuity of control and states $a(t_1^-) = a(t_1^+)$, $v(t_1^-) = v(t_1^+)$, $x(t_1^-) = x(t_1^+)$ and the condition $x(t_1) = x_1$.

Obtaining the state and control trajectories, by solving the above dynamic system, the optimal cost J^* , which now depends on t_1 , is given from the solution of the following optimization problem

$$\begin{aligned} \min J^*(t_1) \\ t_1 \geq t_r \end{aligned} \quad (2.44)$$

The cost function is pseudo-convex for $t_1 \geq t_r$. As a result KKT conditions are both necessary and sufficient and any KKT point is a global minimum.

The KKT conditions are given by

$$\frac{\partial J^*}{\partial t_1} + \mu \frac{\partial (t_r - t_1)}{\partial t_1} = 0 \quad (2.45)$$

$$\mu (t_r - t_1) = 0 \quad (2.46)$$

$$\mu \geq 0 \quad (2.47)$$

The resulting time t_1 is either equal to the value that corresponds to the solution of the above optimization problem without taking into account the inequality constrain ($\mu = 0$), or else it is equal to t_r , due to (2.46) and $\mu = \frac{\partial J^*}{\partial t_1} > 0$. Also in this case it can be seen that, the solution trajectory for the control consists of two linear equations with respect to time t , see (2.41).

2.4.3.2 Discrete-Time Formulation

The discrete-time form of the state equations, for this case with signal constraint, is the same as in (2.38) and the objective function is slightly modified compared to (2.39) by adding a penalty term for the point constraint

$$\begin{aligned} J[v(k), a(k)] = & \frac{1}{2} p_1 [x_e - x(K)]^2 + \frac{1}{2} p_2 [v_e - v(K)]^2 \\ & + \sum_{k=0}^{K-1} \left\{ \frac{1}{2} p_3 \min[0, v(k) + \varepsilon]^2 + f_s[v(k), a(k)] \right\} \\ & + p_4 \sum_{k=0}^{k_1} \left\{ \min(x_1 - x(k), 0)^2 \right\} \end{aligned} \quad (2.48)$$

with the fourth term being the penalty term that prevents position values greater than a given point, x_1 , for a given time t_1 , that corresponds to k_1 , with p_4 being the weight. This penalty term ensures that the vehicle will not pass the traffic light position before switching from red to green.

2.5 Results and Comparisons

In this section, various comparisons are presented to investigate the accuracy and the reliability of the presented variations of the fuel consumption problem. Several scenarios have been tested, with different initial and terminal conditions. In these scenarios, different cases with increasing or decreasing acceleration and deceleration, as well as combinations thereof, were considered (Typaldos, 2017). However, due to limited space, only two scenarios (Scenarios 1 and 2) for

the simple fuel consumption problem, without the signal constraint (Case 1); and one scenario (Scenario 3) for the modified problem (Case 2) will be presented here, with the following initial and final conditions:

Scenario 1. $x_0 = -320$ m, $x_e = 0$ m, $v_0 = 15$ m/s, $v_e = 20$ m/s, $T = 19$ s

Scenario 2. $x_0 = -420$ m, $x_e = 0$ m, $v_0 = 20$ m/s, $v_e = 15$ m/s, $T = 25$ s

Scenario 3. $x_0 = -320$ m, $x_e = 0$ m, $v_0 = 10$ m/s, $v_e = 15$ m/s, $x_1 = -235$ m, $T = 30$ s

Scenario 1 is a positive acceleration scenario where the initial speed is less than the final speed, so the vehicle needs to accelerate in order to reach the final position. On the other hand, Scenario 2 is a negative acceleration scenario with initial speed greater than the final speed which demands from the vehicle to decelerate. As far as Scenario 3 is concerned (traffic light scenario), the initial speed is less than the final speed but, in contrast with Scenario 1, its a mixed acceleration-deceleration scenario as the vehicle has to adjust its speed in order to cross the traffic light after a specific time, t_r , and then fulfill the terminal conditions.

More specifically, the results for Approaches 1 and 1-a, for Case 1, are first derived for Scenarios 1 and 2. Then, the feasible direction algorithm is employed for the solution of Approach 2, for Case 1, comparing different methods of producing descent directions. The results obtained from the different approaches are compared, with respect to the respective optimal control, state trajectories and fuel consumption values, for Scenarios 1 and 2. Subsequently, a comparison of the optimal fuel consumption results of Scenario 1 obtained with different time horizons are presented. Afterwards, the comparison between the optimal trajectories and fuel consumption obtained from Approach 1-a and 2, for Case 2 and for different values of traffic light's switching time, considering Scenario 3, is presented. Finally, the maximization problem for both Cases 1 and 2 using Approach 2 is solved and compared with the corresponding minimization problem.

2.5.1 Solution Approaches 1 and 1-a for Case 1

The resulting optimal trajectories of acceleration, speed and position of the vehicle, for the two mentioned initial and final states Scenarios 1 and 2, are presented in Fig. 2.3 for Approaches 1 and 1-a. The quadratic approximation in Approach 1 was effectuated around the point $(v_k, a_k) = (15$ m/s, 0.05 m/s²), but results would be similar for other points as well. It is evident that the derived results with both approaches are very close to each other; and that the target speed v_e and target position x_e are achieved at the final time T as requested. The fuel consumption values obtained from the ARRB fuel consumption model (2.6) for Scenario 1 are: 42.15 ml for Approach 1, and 42.39 ml for Approach 1-a; while for Scenario 2 they are 28.77 ml for

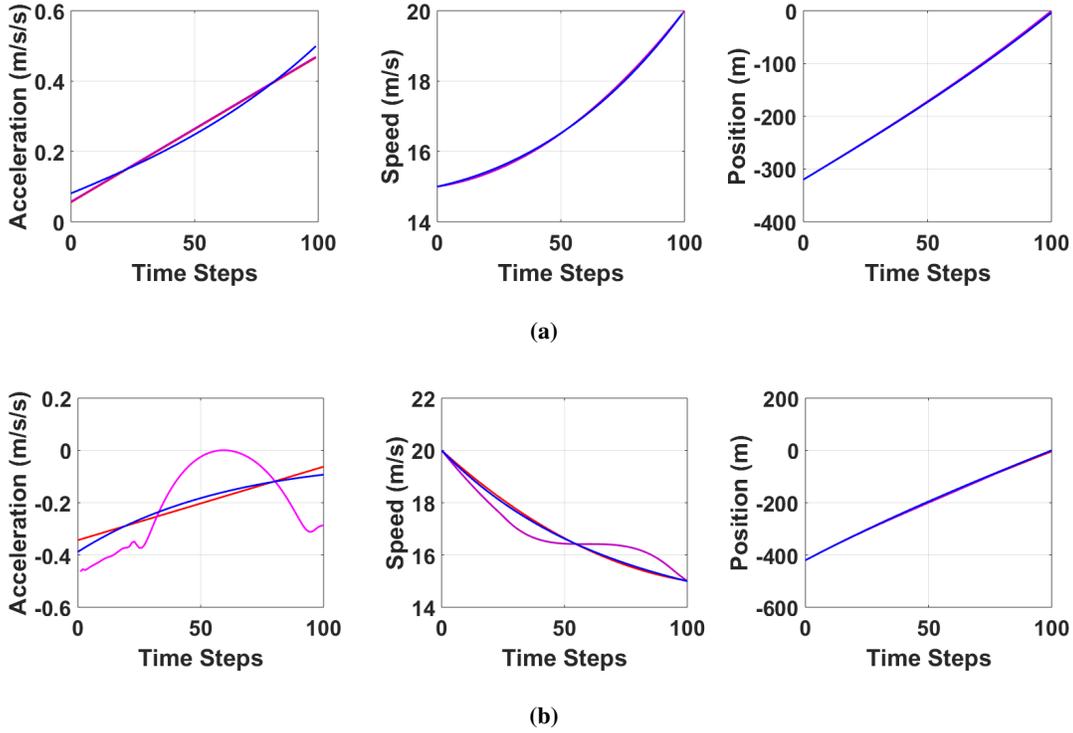


Figure 2.3: Graphical representation of the solution trajectories, for the Approach 1 (blue lines), Approach 1-a (red lines) and Approach 2 (magenta lines), for (a) Scenario 1 and (b) Scenario 2.

Approach 1 and 28.78 ml for Approach 1-a. Note that, for Scenario 1, the vehicle needs to accelerate continuously in order to reach the required final speed of 20 m/s; while in Scenario 2, the vehicle decelerates continuously to reach the final speed of 15 m/s.

2.5.2 Comparison of the Search Direction Algorithms for Approach 2 for Case 1

The performance of different search direction methods used in step 2 of the feasible direction algorithm to produce the numerical solution of Approach 2 for Scenarios 1 and 2 is now investigated. The penalty weights were set to $p_1 = 10^2$, $p_2 = 10^3$ and $p_3 = 10^2$. The discrete-time horizon is set to $K = 100$ with a step length $\Delta t = 0.19$ s for Scenario 1 and $\Delta t = 0.25$ s for Scenario 2. As an initial guess, the trajectory $u(k) = 0$ m/s², $\forall k \in [0, 99]$, was

utilized. The corresponding results of each search direction method are summarized in Table 2.1. From these results, the following conclusions can be drawn:

- All methods converge extremely fast, despite the bad initial guess, to the optimal solution.
- For the addressed problem and settings, all search direction methods, apart from Steepest Descent, are equally efficient.
- The required solution time amounts to only few hundreds of a second.
- In all cases, the final conditions are reached with sufficient accuracy via the quadratic penalty terms.

The evolution of the optimal control and state trajectories $a(k)$, $x(k)$, $v(k)$, for the search direction method DFP are presented in Fig. 2.3. In the following investigations whenever Approach 2 is considered, DFP is utilized.

2.5.3 Comparison of Approaches 1, 1-a and 2 for Case 1

For Scenario 1, it can be observed, from Fig. 2.3a, that the resulting control and state trajectories for all Approaches are virtually identical. The fuel consumption values obtained from the ARRB fuel consumption model (2.6) are: 42.15 ml for Approach 1, 42.39 ml for Approach 1-a and 42.01 ml for Approach 2. Clearly, Approach 2 is always superior, because it directly minimizes the nonlinear (smoothed) ARRB formula, while Approach 1-a minimizes a surrogate (a^2). However, the maximum percentage difference between the fuel consumption values obtained by the different approaches is less than 1%.

On the other hand, for Scenario 2, the trajectories of Approaches 1 and 1-a are almost the same, but the trajectories of the acceleration and speed of Approach 2 differ. This is because, in the deceleration mode, total force R_T is negative, hence the fuel consumption value is constant and equal to the idle fuel consumption. Thus, the fuel consumption value is the same for any deceleration trajectory, the only concern being to eventually reach the required final states. This implies that the acceleration trajectory may take different paths which lead to the same optimal fuel consumption. Indeed, the resulting fuel consumption values are: 28.77 ml for Approach 1, 28.78 ml for Approach 1-a and 28.76 ml for Approach 2; clearly, Approach 2 is again superior, but the percentage difference is less than 0.1%.

¹The results have been acquired with a standard PC with an Intel i5 at 3.2 GHz and 8 GB ram.

Table 2.1: Comparison of search direction methods of Approach 2 for (a) Scenario 1 and (b) Scenario 2

(a)					
Search Direction Algorithm	Iter.	CPU ¹ (s)	$x_1(T)$ (m)	$x_2(T)$ (m/s)	cost (ml)
Steepest Descent	47	0.15	-0.002	19.998	42.01
DFP	4	0.01	-0.002	19.998	42.01
BFGS	6	0.03	-0.002	19.998	42.01
Fletcher-Reeves	4	0.03	-0.002	19.998	42.01
Polak-Ribiere	3	0.02	-0.002	19.998	42.01
(b)					
Search Direction Algorithm	Iter.	CPU ¹ (s)	$x_1(T)$ (m)	$x_2(T)$ (m/s)	cost (ml)
Steepest Descent	23	0.09	-0.004	14.999	28.76
DFP	10	0.04	-0.004	14.999	28.76
BFGS	11	0.05	-0.004	14.999	28.76
Fletcher-Reeves	8	0.04	-0.004	14.999	28.76
Polak-Ribiere	6	0.03	-0.004	14.999	28.76

2.5.4 Approaches 1-a and 2 for Different Time Horizons for Case 1

This section presents the comparison of the optimal fuel consumption values and trajectories obtained from Approaches 1-a and 2, for Scenario 1, for the same time step $\Delta t = 0.19$ s, but different (increased) time horizons. By increasing the time horizon, the scenario is factually modified, because there is more time available to lead the initial states to the final states. Fig. 2.4 displays the control and the states for time steps $K = 100, 200, 300$ and corresponding time horizons $T = 19$ s, $2 \cdot 19$ s, $3 \cdot 19$ s for Approaches 1-a and 2. From Fig. 2.4b and 2.4c, it can be observed that for longer time horizons the problem has been modified into a combined deceleration-acceleration scenario. As the vehicle has more time to reach the final position, it starts by decelerating, in order to achieve the minimum fuel consumption, and accelerates eventually to fulfill the final conditions.

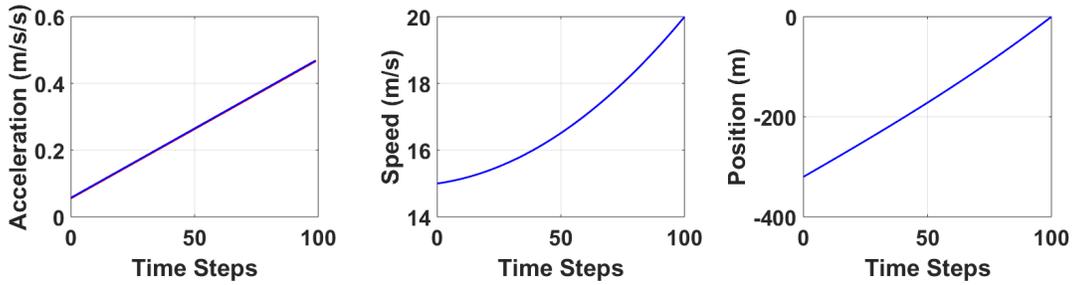
In the results, it can be observed that, when using the initial time horizon (Fig. 2.4a), the

control and states trajectories are virtually identical (as already commented earlier); but as the time horizon is increasing, the acceleration and speed trajectories are slightly differentiated. Specifically, Approach 1-a delivers again a time-linear trajectory for the acceleration; while for Approach 2, as already mentioned, the acceleration has no unique optimal trajectory in the initial phase, due to the constant fuel consumption associated with any deceleration value. For the longest investigated horizon (Fig. 2.4c), we also have activation of the lower speed bound for a short period, which further modifies the acceleration trajectory. Nevertheless, the vehicle position trajectories for both Approaches and all horizons are virtually coinciding, thanks to the smoothing effect of the double integrator applied on the acceleration.

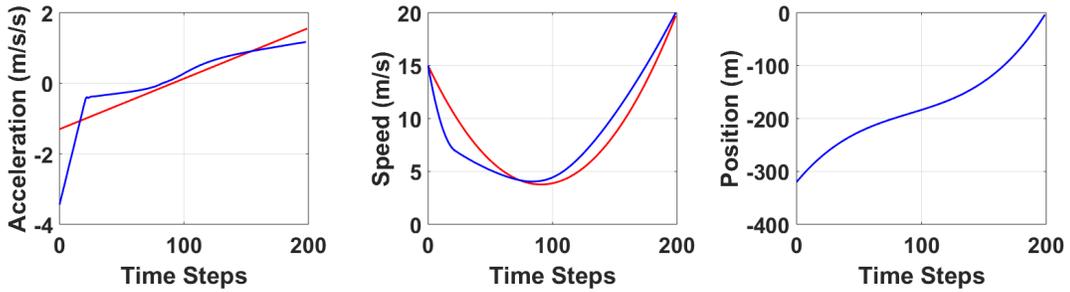
Regarding the percentage difference between the respective fuel consumption values, we have for all the three horizons differences less than 1.3%. Specifically, the fuel consumption values obtained from the ARRB fuel consumption model for $T = 19$ s are: 42.39 ml for Approach 1-a and 42.01 ml for Approach 2; for $T = 38$ s they are: 69.05 ml for Approach 1-a and 68.18 ml for Approach 2; and for $T = 57$ s they are: 82.18 ml for Approach 1-a and 81.34 ml for Approach 2. This outcome supports and underlines the previous conclusion, namely that Approach 1-a is an excellent approximation for fuel minimization, now demonstrated even in scenarios with combined deceleration-acceleration phases of the vehicle.

2.5.5 Approaches 1-a and 2 for Case 2 with Different Stop Point Values

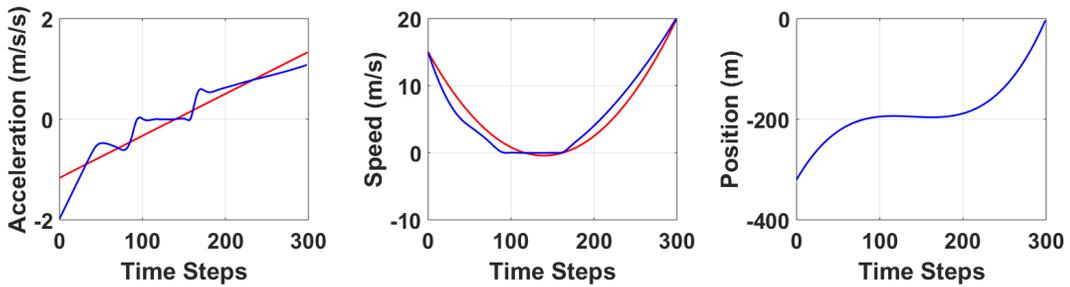
In this section, the optimal trajectories and fuel consumption values obtained from Approaches 1-a and 2, for Case 2 and for different values of time that correspond to k_1 , in which the traffic light switches from red to green, are presented. The corresponding fuel consumption values for each different time (k_1) of traffic light switching are summarized in Table 2.2 for both Approaches 1-a and 2. The resulting graphs of acceleration, speed and position of the vehicle, for Scenario 3, are presented in Fig. 2.5 for Approaches 1-a and 2. It can be observed that, the more the red light period is extended, the more the optimal fuel consumption value is increasing. This behavior is expected for the reason that the vehicle keeps lower speed at the beginning to avoid passing the red traffic light; but then it has to increase, significantly, its acceleration to reach the final position and speed. The trajectories of Case 1 and those of Case 2 with $k_1 = 60$ are visually identical, as for $k_1 \leq 60$ the problem is, essentially, equivalent to that of Case 1. Table 2.2 contains the optimal values of fuel consumption of Approach 1-a and Approach 2, obtained from the ARRB fuel consumption model. It can be observed that, also in this case, the percentage difference of the fuel consumption, obtained from the ARRB model, between Approach 1-a and 2, is less than 1.5%.



(a)



(b)



(c)

Figure 2.4: Control and states for Approach 1-a (red lines) and Approach 2 (blue lines) for the cases of (a) $T = 19$ s and $K = 100$, (b) $T = 38$ s and $K = 200$, (c) $T = 57$ s and $K = 300$.

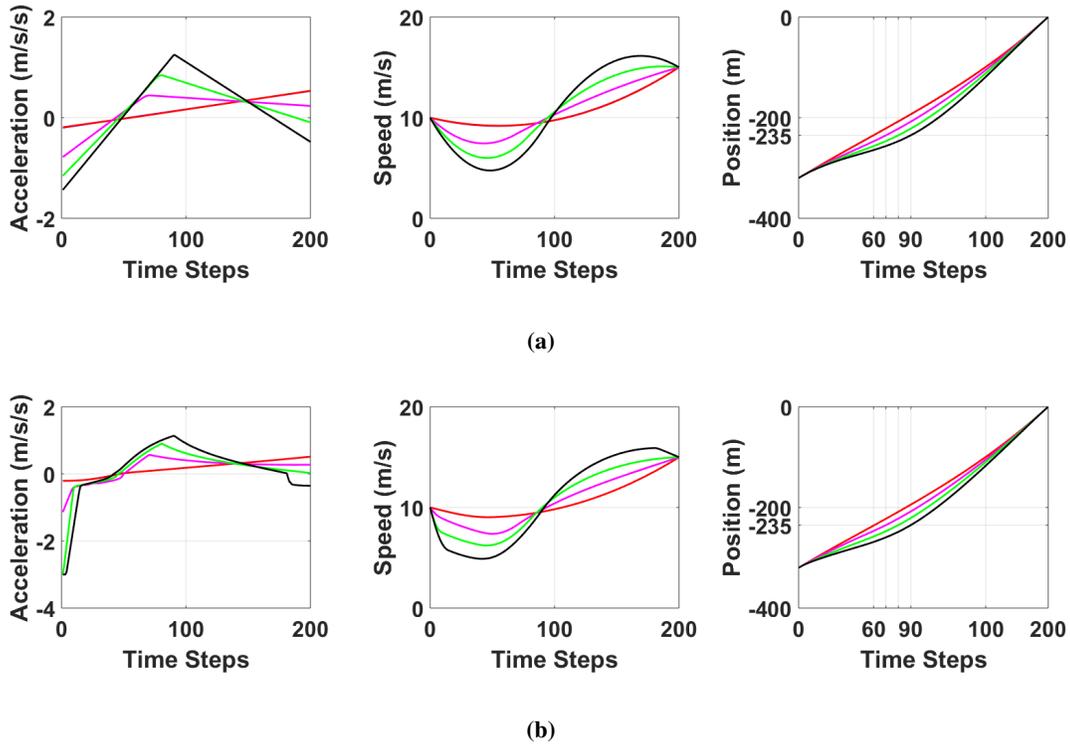


Figure 2.5: Graphical representation of the solution trajectories, for different values of k_1 , where the blue lines are for Case 1 and the red, magenta, green, black lines for Case 2 with $k_1 = 60, 70, 80, 90$, respectively, for (a) Approach 1-a and (b) Approach 2.

2.5.6 Fuel Consumption Maximization Problem for Cases 1 and 2

In the previous investigations, we have demonstrated that the approximate Approaches 1 and 1-a deliver results and fuel consumption values that are within few % from each other. However, one could argue that this might be because the range of possible feasible controls (vehicle maneuvers) that ensure that the vehicle will reach the fixed final state, when started from the given initial state, is accordingly limited. In other words, we would like to investigate what is the range of possible fuel consumption values resulting from feasible control (accelerations). To explore this issue, i.e. to have a feeling of the range of possible maneuvers, we consider in this section the solution of the fuel maximization problem for both Cases 1 and 2 using Approach 2.

Table 2.2: Numerical solutions of Approaches 2 and 1-a for different values of k_1 , using the DFP algorithm. The Cost is the fuel consumption based on the ARRB model.

Cost (ml)	Approach 2	Approach 1-a
Case 1	41.25	41.31
Case 2 with $k_1 = 90$	49.18	49.95
Case 2 with $k_1 = 80$	45.48	45.89
Case 2 with $k_1 = 70$	42.67	42.88
Case 2 with $k_1 = 60$	41.25	41.31

In order to achieve the maximum fuel consumption, maximization problems write the following objective functions need to be solved

$$J[v(k), a(k)] = \frac{1}{2}p_1[x_e - x(K)]^2 + \frac{1}{2}p_2[v_e - v(K)]^2 + \sum_{k=0}^{K-1} \left\{ \frac{1}{2}p_3 \min[0, v(k) + \varepsilon]^2 - f_s[v(k), a(k)] \right\}$$

and

$$J[v(k), a(k)] = \frac{1}{2}p_1[x_e - x(K)]^2 + \frac{1}{2}p_2[v_e - v(K)]^2 + \sum_{k=0}^{K-1} \left\{ \frac{1}{2}p_3 \min[0, v(k) + \varepsilon]^2 - f_s[v(k), a(k)] \right\} + p_4 \sum_{k=0}^{k_1} \left\{ \min[x_1 - x(k), 0]^2 \right\} \quad (2.49)$$

for Cases 1 and 2 respectively, where $f_s[v(k), a(k)]$ is the smoothed nonlinear fuel consumption function (2.10), subject to the same state equations and initial and terminal conditions.

For the maximization problem, an upper and lower bound is applied to the acceleration (see (Papageorgiou et al., 2016)) and also, the penalty term for negative speeds is essential in order to avoid oscillations of the vehicle around the final position with infinite positive and negative speeds.

Fig. 2.6 reflects the comparison between the maximization and minimization problems. Fig. 2.6a displays the optimal state and control trajectories obtained with Approaches 1-a and 2 for Case 1 and Scenario 1, while Fig. 2.6b represents the optimal state and control trajectories

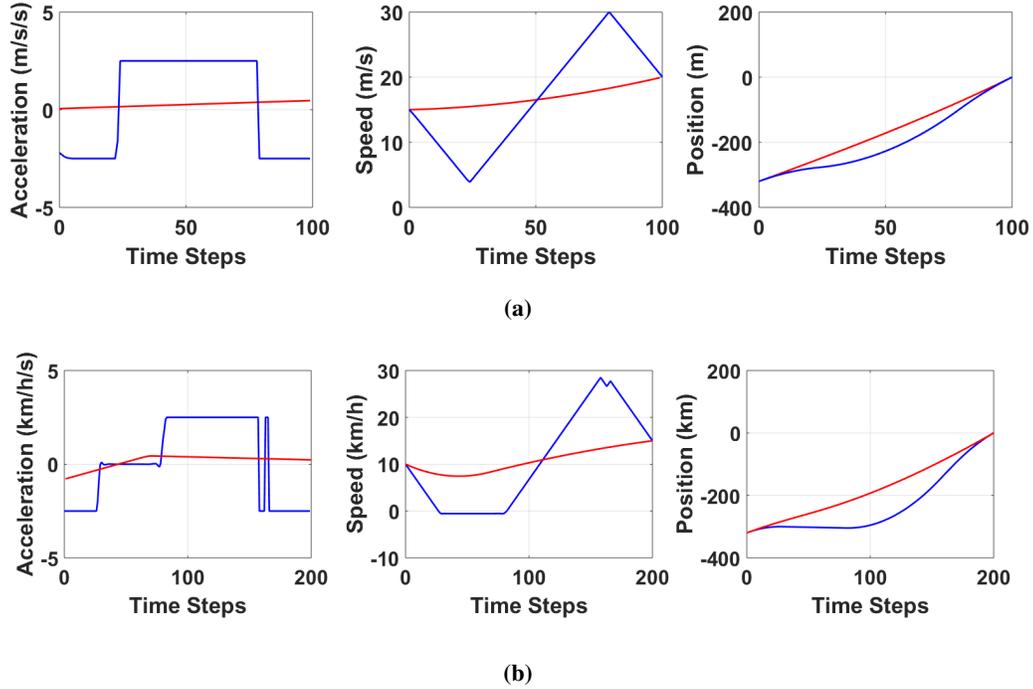


Figure 2.6: Comparison between minimization (red line) and maximization (blue line) problems of the smoothed fuel consumption model, for (a) Case 1 for Scenario 1 and (b) Case 2 for $k_1 = 70$.

of Approach 1-a and 2 for Case 2 and Scenario 3, with traffic light switching time $k_1 = 70$. The acceleration bound have been set to $a_{\min} = -2.5 \text{ m/s}^2$ and $a_{\max} = 2.5 \text{ m/s}^2$.

The fuel consumption values resulting from the two maximization problems are 138.89 ml and 145.29 ml, respectively; while for the minimization cases they were 42.39 ml and 42.88 ml, respectively. The percentage difference between the two solutions for both cases is more than 220%, which indicates that the range of possible fuel consumption values while driving the vehicle from the given initial state to the given final state is indeed substantial, and, hence, the previously achieved approximations with Approach 1 and 1-a are indeed excellent.

From Fig. 2.6, a bang-bang behavior can be observed between the minimum and maximum values of the control, a_{\min} and a_{\max} , for the maximization case. This behavior is expected, as the vehicle has to reach high acceleration values in order to achieve the maximum fuel consumption and, eventually, it has also to reach high values of deceleration in order to fulfill the final state conditions of the problem.

2.6 Conclusion

The focus of this work is the minimization of fuel consumption of a vehicle, by optimizing its kinematic trajectories. The fuel consumption problem was formulated as an optimal control problem and was solved, firstly, analytically (after an appropriate approximation (Approach 1 and Approach 1-a)) and then numerically (Approach 2). The results obtained from all the solutions were very satisfactory, and the percentage difference of fuel consumption between all the Approaches was negligible. Regarding the comparison of the resulted trajectories, for small time horizon, Approaches 1-a and 2 give almost the same trajectories, but, as the time horizon are increased, the trajectories increasingly differ, but the percentage difference of the fuel consumption values remain very low (less than 1.3%).

Finally, the maximization problem for the fuel consumption was also solved, and the obtained results are compared with those of the minimization problem. Through these results, the value range of the fuel consumption model was investigated and, with the percentage difference of the two solutions being large (220% in the scenario examined) it can be concluded that the presented simplified fuel consumption approaches are highly efficient.

In summary, this work has delivered two main contributions:

- It proposed a highly efficient (fractions of a second) numerical approach for the solution of the accurate fuel minimization problem (Approach 2).
- It demonstrated in a variety of ways that simpler approaches (Approach 1-a) deliver excellent approximations of the accurate solution.

Acknowledgment

The research leading to these results has received funding partially from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. [321132], project TRAMAN21 and partially from the European Commission under the European Union's Seventh Framework Programme (FP/2007-2013) / FP7-ICT-2013.3.4, project LOCAL4GLOBAL (n. 611538).

Chapter 3

Optimization-Based Path-Planning for Connected and non-Connected Automated Vehicles

3.1 Abstract

A path-planning algorithm for connected and non-connected automated road vehicles on multilane motorways is derived from the opportune formulation of an optimal control problem. In this framework, the objective function to be minimized contains appropriate respective terms to reflect: the goals of vehicle advancement; passenger comfort; and avoidance of collisions with other vehicles and of road departures. Connectivity implies, within the present work, that connected vehicles can exchange with each other (V2V) real-time information about their last generated short-term path. For the numerical solution of the optimal control problem, an efficient feasible direction algorithm (FDA) is used. To ensure high-quality local minima, a simplified Dynamic Programming (DP) algorithm is also conceived to deliver the initial guess trajectory for the start of the FDA iterations. Thanks to very low computation times, the approach is readily executable within a model predictive control (MPC) framework. The proposed MPC-based approach is embedded within the Aimsun microsimulation platform, which enables the evaluation of a plethora of realistic vehicle driving and advancement scenarios under different vehicles mixes. Results obtained on a multilane motorway stretch indicate higher efficiency of the optimally controlled vehicles in driving closer to their desired speed, compared to ordinary manually driven vehicles. Increased penetration rates of automated vehicles are found to increase the efficiency of the overall traffic flow, benefiting manual vehicles as well. Moreover, connected controlled vehicles appear to be more efficient in achieving their desired speed, compared also to the corresponding non-connected controlled vehicles, due to the improved real-time information

and short-term prediction achieved via V2V communication.

3.2 Introduction

In the past decade, automated driving has attracted strong interest in industry and scientific community. This is fostered by strong technological advancements, compared to which human driving capabilities appear limited in terms of perception of the driving environment, reaction time, and real-time decision efficiency. In addition, variations in driving behavior from person to person or short inattention at high speeds may result in accidents. In fact, the vast majority of road accidents are attributed to human error on the account of, e.g., insufficient sensory information, lack of attention, shortcomings in driving skill or reckless driving. Each year, road accidents result in approximately 1.35 million fatalities and leave some 50 million of injured or disabled worldwide. Road congestion is another major issue, causing excessive delays, fuel consumption and emissions around the globe (Goniewicz et al., 2016; Montanaro et al., 2019). On the other hand, vehicle automation is a challenging area due to the variety and complexity of real-world environments, including avoidance of static and moving obstacles, compliance with traffic rules and consideration of human driving behavior aspects (Gu, Dolan, 2014).

With the recent advances in vehicle communications, either vehicle-to-vehicle (V2V) or infrastructure-to-vehicle (I2V), new communication channels, carrying a potentially wide range of information becomes available in real time. Connected automated vehicles (CAVs) may receive or exchange relevant information, including vehicle state, the current traffic conditions, the next switching time of a traffic signals etc. This extended information entails better knowledge of the driving conditions for CAVs and may be beneficial in various driving situations concerning road safety, flow efficiency and environmental sustainability (Sjoberg et al., 2017; Tian et al., 2018). Specifically, automated driving may improve various driving aspects, such as lane-changing, obstacle avoidance, forming of platoons with short inter-vehicle distances and the application of smoother acceleration or deceleration.

The development of fully automated driving algorithms is inherently related to planning and updating a vehicle path, which should be efficient, collision-free and user-acceptable. Planning such a path can be viewed as a trajectory generation problem, i.e., creation in real time of a quasi-continuous sequence of states that must be tracked by the vehicle via appropriate steering, throttle and braking actions. Inspired by earlier studies on motion planning of robot vehicles in other contexts (e.g. (González et al., 2015)) and driven by imminent implementations and optimistic forecasts of CAV technologies (e.g. (Gu, Hu, 2002; Nagy, Kelly, 2001)), studies on CAV trajectory optimization in the road traffic context have boomed in the past decade. However, path-planning for road vehicles is a difficult task, since speeds are very high, and safety of the passengers must be guaranteed. Thus, automation on

roads calls for sophisticated approaches, such as optimal control methods, advanced feedback control or reinforcement learning (Claussmann et al., 2019; Haydari, Yilmaz, 2020).

In the related literature, many works exist for trajectory generation of longitudinal motion for CAVs, considering cooperative adaptive cruise control (CACC) and platooning (Dey et al., 2015; Wang et al., 2018b); cooperative merging at highway on-ramps (Ntousakis et al., 2016; Rios-Torres, Malikopoulos, 2016a); speed harmonization on highways (Ghiasi et al., 2017; Malikopoulos et al., 2018b) etc. On the other hand, there are relatively few studies on CAV trajectory optimization considering lane-changing (Wang et al., 2018c), which indicates that optimization of CAV trajectories in both longitudinal and lateral directions is worth additional investigations. Some of these studies approach the problem with optimal control and model predictive control (MPC), while avoidance of collision with other vehicles is handled through potential-field like penalty functions (Dixit et al., 2019; Jalalmaab et al., 2015; Makantasis, Papageorgiou, 2018; Rasekhipour et al., 2016; Wang et al., 2018a). In comparison with other aforementioned approaches, e.g. reinforcement learning, optimal control techniques require substantially less data as input. Moreover, depending on the numerical solution algorithms, the execution times of the problem should be sufficiently small to enable their implementation within a real-time MPC framework, i.e. real-time feasibility.

As far as the connectivity capabilities of the CAVs is concerned, most of the related mentioned works consider limited knowledge of their surroundings, specifically knowledge of the current position and speed of surrounding vehicles, e.g. through their own sensors. Thus, the short-term future path of the obstacles is considered as a projection of their initial states, e.g. assuming zero acceleration and no lane change. For example, in (Sadat et al., 2020; Shao et al., 2021), the prediction of the surrounding environment is handled through marginal distributions of semantic occupancy over time; or considered as static region, respectively. In the same context, (Saxena et al., 2020) studies the problem of successfully executing a safe and comfortable merge into dense traffic, while the surrounding obstacle vehicles are handled either as static or as constant-speed obstacles. Note that a variety of path planning approaches that emerged in decade-long research in the Robotics domain are valuable for road vehicle path planning as well, but there is a crucial difference in the respective requirements due to automated highway vehicles (and dynamic obstacles) driving much faster than typically considered autonomous robots. This important difference calls for enhanced prediction of the obstacles behaviour, which is indeed achieved via vehicle connectivity, as considered in this work.

Compared to approaches presented in the past by some of the authors (Makantasis, Papageorgiou, 2018), the present work marks significant differences and advances. In that early-stage work, a basic optimal control problem (OCP) formulation was developed, providing a solid basis for subsequent significant advances, both in the conceptual design and in its evaluation. Specifically, significant differences in the present work include: the newly

designed collision avoidance term, the number of involved state variables, the prediction of the surrounding vehicles' paths and the comprehensive and realistic evaluation at traffic level, to mention just the most important ones. In more detail, in the previous work, the collision avoidance term was formulated as a non-smooth function, which may entail difficulties for gradient-based numerical solution algorithms, such as the employed feasible direction algorithm (FDA). Therefore, in this work, the corresponding term has been modified, and a novel smooth function is appropriately designed. In addition, the longitudinal control variable in this work is the jerk, instead of the acceleration, which ensures smoother trajectories and improves the passengers' comfort. Moreover, the specific previous work did not consider connectivity for the vehicles, which implies that the prediction of the surrounding vehicle paths has limited accuracy, and the impact of vehicle connectivity cannot be assessed. Finally, compared to the previous work, which only reports on limited open-loop evaluation instances involving just a few vehicles, without testing its applicability within a closed-loop MPC framework, this work develops the proposed approach fully, along with the appropriate MPC framework; and implements it within a realistic microsimulation platform (Aimsun Next, 2019) in order to comprehensively evaluate the traffic-level effects of the controlled AVs at various demand levels and for different penetration rates of vehicle automation and connectivity.

Compared to other works in the literature, the proposed approach delivers a very efficient algorithm, which is readily applicable in real time due to very low computational times (fractions of a second). Moreover, the introduction of the enhanced information of the CAVs enables more accurate prediction of their surrounding and is demonstrated to be crucial, particularly in scenarios of high demand levels. Finally, based on the authors' knowledge, there are no comparable works considering such a comprehensive OCP and MPC-based approach within a microscopic simulator for various demand levels and, especially, for various penetration rates; and reporting on the effects of the AVs on their own efficiency and on the traffic environment, as the penetration rate rises.

In summary, the main contributions of this work are the following:

- A highly efficient (fractions of a second) numerical approach for the solution of the OCP for path planning of AVs.
- The full exploitation of vehicle connectivity via asynchronous path decision exchange that leads to more accurate prediction of the surrounding CAV behaviour.
- Comprehensive traffic-level evaluation and analysis, including the impact of lane changes, for different penetration rates and demand levels.

In more detail, this paper explores the impact of vehicle connectivity on the efficiency of advancement of automated vehicles (AVs) and on the efficiency of the emerging traffic flow.

The investigations address multi-lane motorways with mixed traffic, comprising both automated and manually driven vehicles at different penetration rates. An optimal control problem is employed for the path-planning of AVs, comprising three main elements:

- A simple kinematic model describing the vehicle movement process.
- An objective function to be minimized, which contains respective terms to reflect efficient vehicle advancement; passenger comfort and fuel consumption; and avoidance of collisions with other vehicles and of road departures.
- Short-term prediction of the trajectories of other neighboring vehicles (obstacles), which is crucial for pro-active collision avoidance.

For the numerical solution of the optimal control problem (OCP), a very efficient iterative feasible direction algorithm (FDA) is used. To ensure high-quality local minima, a simplified Dynamic Programming (DP) algorithm is also employed to deliver the initial guess trajectory for the FDA. Thanks to low computation times, the overall approach is readily executable within a MPC framework, applying updated initial state and obstacle path prediction at each repetition. MPC has a long history of applications in control, automation, and chemical industries (Mayne, 2014; Mayne, Michalska, 1988; Qin, Badgwell, 2003). Due to the strongly dynamic environment of the path-planning problem for AVs, the accurate prediction of the surrounding vehicles' paths is crucial, particularly at high highway speeds. However, the uncertainties related to the changing environment and to the actual vehicle advancement inevitably increase over time. Therefore, the MPC approach is often used for online path planning of AVs, as it is very efficient in handling the arising uncertainties due to its re-planning ability (Claussmann et al., 2019). One possibility is to solve the OCP at each time step, with a corresponding shift of the planning horizon (Howard et al., 2010; Murillo et al., 2018; Nilsson et al., 2015). An alternative possibility is to update path decisions in event-based mode (Earl, D'Andrea, 2007; Khazaeni, Cassandras, 2016), which is pursued in the current study.

Vehicle connectivity refers here to the capability of AVs to exchange with each other, in an asynchronous mode, real-time information about their current state (position and speed) and their latest generated path, where the latter strongly enhances the prediction accuracy for obstacle movement in the short-term future. The MPC-based approach is embedded within the Aimsun micro-simulation platform (Aimsun Next, 2019), which enables driving evaluation in countless appearing driving scenarios. Specifically, we investigate two cases, each of them at different penetrations of AVs:

- No connectivity: Each AV is aware of the current position and speed of obstacles (via its own sensors). Short-term prediction of obstacle movement is based on extrapolation, assuming zero acceleration and same-lane movement.
- Connected automated vehicles: Each AV is aware of the current position and speed of obstacles; in addition, it receives the path-planning decisions of other automated vehicles, which facilitates the short-term prediction of their movement.

In both cases, the short-term movement prediction for manually driven vehicles is based on zero-acceleration and same-lane extrapolation.

Demonstration results are reported for a motorway pipeline section. The results indicate higher efficiency of the optimally controlled vehicles in driving closer to their desired speed, compared to non-automated vehicles. In addition, increasing penetration rates of controlled vehicles are found to lead to more efficient traffic flow. Specifically, as the penetration rate of controlled vehicles rises, there is a significant increase of the average speed and, consequently, a decrease on the average delay time and travel time for all vehicles. These conclusions apply mainly for demand levels where the controlled vehicles have space to manoeuvre. Specifically, the proposed approach is efficient due to the fact that the controlled vehicles are able to apply smarter manoeuvres. Also, connected controlled vehicles are more efficient than non-connected controlled vehicles, due to the improved real-time information that enables more pertinent obstacle movement prediction. Finally, the emerging traffic flow and even the manually driven vehicles are found to benefit from improved operation of the AVs.

It should be noted that urban road environments are out of the scope of this work, which focuses on vehicles moving on a highway. We believe that application of a similar approach for urban networks is possible with appropriate modifications or extensions, but this is left to future work.

The rest of the paper is organized as follows: Section 3.3 presents the dynamics of each AV and the components of the objective function that lead to the OCP definition. Section 3.4, describes the numerical solution algorithms and explains the procedure used for MPC. Section 3.5 presents the simulation results, while Section 3.6 concludes this work.

3.3 Optimal Control Problem Formulation

This section describes the proposed path-planning strategy for automated road vehicles on motorways. At first, the simple vehicle kinematic motion dynamics and considered hard constraints are defined. Then, an objective function is designed, which includes appropriate terms regarding the efficient vehicle advancement, the obstacle and off-road avoidance and the

passenger convenience. Finally, the proposed path-planning algorithm is generated based on a combination of FDA and Dynamic Programming techniques.

3.3.1 Problem Variables, State-Equations and Constraints

We consider a straight road on a two-dimensional plane, and the vehicle's position on this plane is expressed in global Euclidean coordinates. Each controlled vehicle is described by five state equations, corresponding to the equations of motion, which are expressed in discrete time, assuming time-steps of length T , as follows:

$$x(k+1) = x(k) + v_x(k) \cdot T + \frac{1}{2} a_x(k) \cdot T^2 + \frac{1}{6} j_x(k) \cdot T^3 \quad (3.1)$$

$$y(k+1) = y(k) + v_y(k) \cdot T + \frac{1}{2} a_y(k) \cdot T^2 \quad (3.2)$$

$$v_x(k+1) = v_x(k) + a_x(k) \cdot T + \frac{1}{2} j_x(k) \cdot T^2 \quad (3.3)$$

$$v_y(k+1) = v_y(k) + a_y(k) \cdot T \quad (3.4)$$

$$a_x(k+1) = a_x(k) + j_x(k) \cdot T \quad (3.5)$$

where $x(k), y(k), v_x(k), v_y(k), a_x(k)$ correspond to the longitudinal and lateral position, the longitudinal and lateral speed and the longitudinal acceleration at time-step k , respectively; while the control variables $j_x(k), a_y(k)$ refer to the longitudinal jerk and the lateral acceleration, respectively, which are kept constant for the duration T of each time-step k ; hence the above state equations are derived from the exact time-integration of the corresponding continuous-time differential equations of motion. Note that the consideration of jerk (rather than the acceleration) as a control variable for the longitudinal direction leads to smoother vehicle trajectories, which consequently improve the convenience of the vehicle passengers. On the other hand, for the lateral movement, such detail is not necessary, as the lateral speed and movement is only needed for lane changing, and a lane change maneuver is substantially less frequent compared to the continuous longitudinal motion. It should be noted that, in contrast to other works, the proposed approach produces continuous, rather than discrete, lateral movement, thus avoiding the usage of integer variables that increase the computational effort for the numerical OCP solution.

In this work, the model for expressing the vehicle dynamics is a double (lateral) and triple (longitudinal) integration model, which is quite common for the problem at hand, involving high-speed longitudinal movement with occasional lane changes. This model considers the lateral and longitudinal movement to be decoupled. An alternative would be the use of a so-called bicycle model (Rajamani, 2011). However, in a lane-based highway environment, a

vehicle mostly follows a leader or drives on its lane with its desired speed, as the lane-change manoeuvres are not very frequent. In these conditions, explicit consideration of steering is not deemed important.

The control variables $j_x(k), a_y(k)$ are bounded according to vehicle specifications, whereby the longitudinal upper bound and the lateral bounds are constant, while the longitudinal lower bound is state-dependent, as follows:

$$j_{x,\min}(v_x(k), a_x(k)) \leq j_x(k) \leq j_{x,\max} \quad (3.6)$$

$$a_{y,\min} \leq a_y(k) \leq a_{y,\max} \quad (3.7)$$

The constant upper bound $j_{x,\max}$ and the lateral bounds $a_{y,\min}, a_{y,\max}$ may be set with consideration of the vehicle capabilities and the passengers' convenience. Furthermore, a vehicle, having a currently non-negative longitudinal speed $v_x(k)$, should not have negative longitudinal speed at the next time step, i.e.,

$$v_x(k+1) \geq 0 \quad (3.8)$$

should hold. To ensure this, while avoiding state constraints that may complicate the numerical solution of the OCP, the state equation (3.3) is replaced in (3.8) and, after rearrangement, this yields the state-dependent control constraint

$$j_x(k) \geq -\frac{2}{T^2}v_x(k) - \frac{2}{T}a_x(k) \quad (3.9)$$

This bound may be unrealistically low (negative) at higher speeds, due to the magnitude of the coefficients, which can be a cause of discomfort for the passengers. Note that the equation-version of (3.9) may be interpreted as a dead-beat controller that drives the speed v_x to zero in exactly two time steps. However, the magnitude of the resulting jerk bounds can be mitigated by choosing more moderate "controller" coefficients $0 < K_1 \leq 2/T^2$ and $0 < K_2 \leq 2/T$, which would drive v_x to zero asymptotically. This way, accordingly moderate lower jerk bounds are obtained, while guaranteeing that the constraint (3.8) is always satisfied. In conclusion, the considered lower bound on longitudinal jerk is

$$j_{x,\min}(v_x(k), a_x(k)) = -K_1 \cdot v_x(k) - K_2 \cdot a_x(k) \quad (3.10)$$

and (3.6) is replaced by the following state-dependent constraint to be considered in OCP

$$h = [j_x(k) - j_{x,\max}][j_x(k) - j_{x,\min}] \leq 0 \quad (3.11)$$

3.3.2 Optimization Objectives

3.3.2.1 Objective Function

The objective function to be minimized, in the frame of the OCP formulation, includes a number of terms, which consider efficient, convenient and safe driving, each with a weighting parameter to reflect the corresponding priorities. Among these terms, several safety constraints are considered as soft constraints, so as to drastically speed up the numerical solution. Although alternative approaches may indeed lead to respect of the safety constraints in a more direct way, our indirect approach (using soft constraints) also satisfies these constraints and fulfils this task to the full, while featuring very low CPU times, which make it readily applicable in real time.

The optimisation criterion, which extends over a time horizon of K steps in the future, reads

$$J = \sum_{k=0}^{K-1} \left[w_1 j_x^2(k) + w_2 a_x^2(k) + w_3 a_y^2(k) + w_4 [v_x(k) - v_{d,x}]^2 \right. \\ \left. + w_5 [v_y(k) - v_{d,y}]^2 + w_6 f_r(y(k)) + w_7 \sum_{i=1}^n [c_i(x(k), y(k))] \right] \quad (3.12)$$

where w_1, \dots, w_7 are non-negative penalty parameters.

The first three quadratic penalty terms concern the comfort of the passengers, which is related to the magnitude of lateral and longitudinal acceleration, as well as of the longitudinal jerk. Note that the quadratic penalty term of longitudinal acceleration acts also as a good proxy for deriving fuel-minimizing vehicle trajectories (Typaldos et al., 2020b). More specifically, it has been demonstrated that the simple square-of-acceleration term delivers excellent approximation of fuel-optimal vehicle trajectories, when compared with the use of complex and realistic fuel consumption models in the objective function.

The fourth and fifth penalty terms reflect the vehicle advancing goals. These terms account for pre-specified desired longitudinal and lateral speeds by penalizing speed deviations from those values. In the current work, $v_{d,x}$ has a positive value, corresponding to the vehicle type or driver choice of desired longitudinal speed; while $v_{d,y}$ is set to zero to suppress unnecessary lateral movements. Note that, $v_{d,y}$ can be set to non-zero values in case, for example, a vehicle is bound to exit the motorway at a downstream off-ramp or in emergency cases, where the controlled vehicles should create appropriate space, e.g., for an ambulance or a fire truck to pass.

The two last penalty terms, which are related to the lateral road boundaries and the obstacle avoidance, respectively, are described in detail in the following.

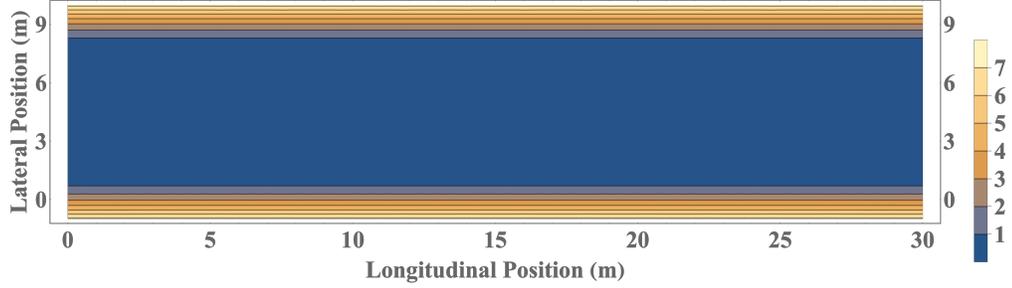


Figure 3.1: The penalty function of road boundaries.

3.3.2.2 Road Boundaries Term

The penalty function for the road boundaries must be designed so as to disallow the automated vehicle (AV) to depart from the road. To this end, the following smooth quadratic function, dependent on the width w of the road, is adopted, which is equal to zero when the vehicle moves within the road boundaries; while its value increases the more the vehicle would depart from the boundaries

$$f_r = \begin{cases} (y - d)^2 & , y < d \\ 0 & , d \leq y \leq w - d \\ (y + d - w)^2 & , y > w - d \end{cases} \quad (3.13)$$

where the small positive parameter d is used to prevent the ego vehicle from moving too close to the road boundaries; and w is the road width. Fig. 3.1 displays the penalty function generated for the road boundaries for a three-lane motorway with $w = 9$ m and $d = 1.5$ m.

3.3.2.3 Collision Avoidance Term

Crash avoidance is an absolute necessity in vehicle path planning, that may be achieved in different ways. Note that crash avoidance is closely related to the highest deceleration available, not just the way (hard or soft) of considering the related constraints. Thus, it is crucial that deceleration vis-à-vis an obstacle is applied timely, so as to avoid the need for harsh or infeasible deceleration eventually; and this is indeed a feature enabled via OCP with future time horizon (that helps anticipate imminent crash risks) and soft constraints (that exert timely smooth repulsion while the controlled vehicle is approaching obstacles).

The penalty function (similar to potential field functions) for obstacle avoidance should feature high values at the gross obstacle space, so that the ego vehicle is repulsed, and potentially unsafe trajectories are suppressed; and low (or virtually vanishing) values outside

of that space. To this end, we adopt, for each obstacle, an ellipsoid penalty function, which is best understood in its one-dimensional form $f(x) = (1 + (x/a)^{p_1})^{-1}$. This smooth function has a maximum of 1 for $x = 0$; for $|x| < a$, the function retains values close to 1 (the maximum), while for $|x| > a$, it reduces to very small values or virtually zero. The even integer parameter p_1 influences the sharpness of the smooth transition of function values from 1 (for $|x| < a$) and 0 (for $|x| > a$).

For the present needs, the function is generalised to two dimensions, and its two respective arguments δ_x and δ_y reflect the longitudinal and lateral distances between the ego vehicle and each obstacle i , while the respective counterparts of parameter a should be selected based on the vehicle dimensions, taking into account also the vehicle and obstacle speeds for safe car-following distances. Thus, in two dimensions, the following ellipsoid penalty function for each obstacle i is used

$$c_i(x, y) = \frac{1}{\left(\frac{\delta_x + s}{0.5 \cdot r_x}\right)^{p_1} + \left(\frac{\delta_y}{0.5 \cdot r_y}\right)^{p_2} + 1} \quad (3.14)$$

where p_1 and p_2 are positive even integers, $\delta_x = x - x_i$ and $\delta_y = y - y_i$ are the longitudinal and lateral distances from obstacle i , r_x and r_y determine the dimensions of the ellipse, and the term s introduces a shift of the ellipse's longitudinal centre position, as will be explained in the following.

For safety, the width of the ellipse, r_y , should be equal to the width of a motorway lane, W , to avoid lateral intrusion of the ego vehicle into a lane occupied by an obstacle. On the other hand, the length of the ellipse, r_x , should consider, beyond the physical vehicle length, a safe space gap in front and behind of each obstacle i . In case the ego vehicle is behind the obstacle, a safe space-gap, equal to $\omega \cdot v_x$, should be maintained between the two vehicles, where ω is the time-gap parameter (Rajamani, 2011). On the other hand, when the ego vehicle is in front of the obstacle, a safe space-gap, equal to $\omega \cdot v_i$, should be considered, with v_i being the longitudinal speed of the obstacle. Moreover, the physical dimensions of both ego and obstacle vehicles, $L_i = (\mu_e + \mu_{o,i})/2$, with μ_e , $\mu_{o,i}$ being the ego and obstacle's i lengths, respectively, should be considered as a minimum safe space-gap in case of zero speeds. Thus, the longitudinal ellipsoid dimension is defined as

$$r_x = \omega \cdot v_x + \omega \cdot v_i + L_i \quad (3.15)$$

Due to the difference in the longitudinal speeds of the ego vehicle and the obstacle, the above space gaps in front of and behind the obstacle are accordingly different, hence the ellipse is longitudinally asymmetric with respect to the physical centre of the obstacle. Therefore, the ellipse centre must be appropriately shifted, depending on ego vehicle and obstacle speeds difference. Specifically, the position of the ellipse centre is expressed as the summation of δ_x

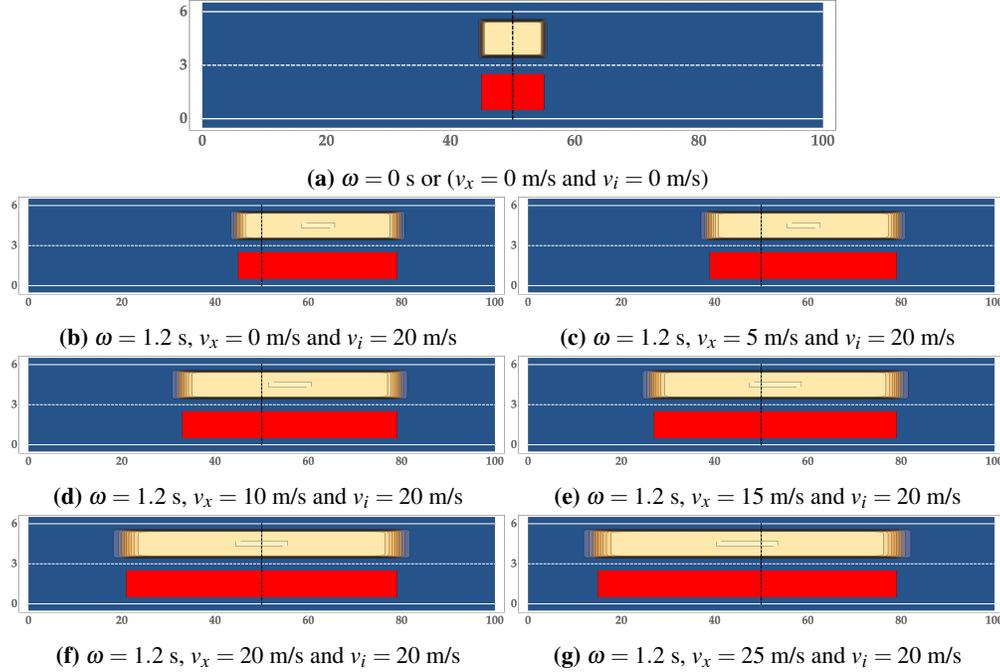


Figure 3.2: Illustration of the collision avoidance term (orange ellipsoid), $c_i(x,y)$, for different ego vehicle speeds with time gap $\omega = 1.2$ s and obstacle speed $v_i = 20$ m/s, compared to the corresponding space gap of the constant time gap policy (red rectangle). The lateral dashed line indicates the physical vehicle centre location.

and the shift term s given by

$$s = \frac{\omega \cdot (v_x - v_i)}{2} \quad (3.16)$$

For safe car-following, the longitudinal size of the ellipse depends on the controlled vehicle's and obstacle's speeds, as well as the time-gap value employed. Thus, the size is not fixed, but changes over time according to the mentioned variables. For example, when the vehicles are stopped (zero speed), the ellipse's longitudinal dimension equals the average length of the controlled vehicle and the leading vehicle, in both sides of the ellipse; while, as the speed of both vehicles increases, the dimension of the ellipse increases accordingly.

In Figs. 3.2a-3.2g, different cases for the proposed ellipsoid are displayed for illustration. We assume an obstacle with its centre point at 50 m. The ellipsoids in each case are displayed in orange color and are contrasted to the corresponding space gaps resulting from the constant

time-gap policy (red rectangle) (Rajamani, 2011). In Fig. 3.2a, both the ego vehicle and the obstacle have zero speed, leading to a symmetric ellipse shape around the obstacle centre, as both the front and the back part around the vehicle centre (dashed black line) are equal to L_i . Note that the shape of the ellipse is close to a rectangle due to the use of high values for both exponents p_1 and p_2 in (8) (e.g. $p_1 = p_2 = 18$). In Figs. 3.2b-3.2g, using a time-gap value $\omega = 1.2$ s and a longitudinal obstacle speed $v_i = 20$ m/s, it is observed how the ellipse shape changes, and the ellipse centre shifts in dependence of the ego vehicle speed. Specifically, when the ego vehicle has zero speed (3.2b), the centre of the ellipse is shifted to upstream to accommodate a back space-gap equal to the average length of both vehicles; while the front part of the ellipse equals the space-gap, which depends on the speed of the obstacle. For higher ego vehicle speeds, it can be observed that the length of the ellipse increases, and, at the same time, its centre shifts downstream. In the specific representation, the length of the front part of the ellipse does not change, as the obstacle speed remains constant.

Potential obstacles, to be considered in the sum of the collision avoidance term in (3.12), are all vehicles around the ego vehicle that might appear on its way, causing potentially a crash risk, during the considered planning horizon. Thus, all vehicles within a longitudinal zone in front and behind the ego vehicle should be considered as potential obstacles, and the length of this zone is taken proportional to the ego vehicle's desired longitudinal speed times the time horizon.

3.3.3 Problem Formulation

In conclusion, the path-planning problem may be formulated as OCP. The difference equations (3.1)-(3.5) may be organised in the following vector form

$$\mathbf{x}(k+1) = \mathbf{f}[\mathbf{x}(k), \mathbf{u}(k), k], \quad k = 0, \dots, K-1 \quad (3.17)$$

where \mathbf{x} and \mathbf{u} are the system states and control variable vectors, respectively. With known initial state $\mathbf{x}(0) = \mathbf{x}_0$ and obstacle trajectories, the optimal control problem consists in minimizing the objective function (3.12) subject to (3.17) and control bounds

$$\mathbf{h} = [\mathbf{u}(k) - \mathbf{u}_{\max}][\mathbf{u}(k) - \mathbf{u}_{\min}] \leq \mathbf{0}, \quad k = 0, \dots, K-1 \quad (3.18)$$

which reflect constraints (3.7) and (3.11) in a unified way.

The general form of the objective function (3.12) is given by

$$J = \sum_{k=0}^{K-1} \phi[\mathbf{x}(k), \mathbf{u}(k)] \quad (3.19)$$

and then, the Hamiltonian function is given by (see (Papageorgiou et al., 2015, 2016))

$$H[\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\lambda}(k+1), k] = \phi[\mathbf{x}(k), \mathbf{u}(k)] + \boldsymbol{\lambda}(k+1)^T f[\mathbf{x}(k), \mathbf{u}(k), k] + \boldsymbol{\mu}(k)^T \mathbf{h}[\mathbf{x}(k), \mathbf{u}(k)] \quad (3.20)$$

where $\boldsymbol{\lambda}(k+1)$ stands for the co-state vector (equivalent to Lagrange multipliers) for the corresponding state equations.

The following conditions of optimality for the discrete-time OCP must be satisfied for $k = 0, \dots, K-1$ (notation: $z_w = \partial z / \partial w$)

$$\mathbf{x}(k+1) = H_{\boldsymbol{\lambda}(k+1)} = f[\mathbf{x}(k), \mathbf{u}(k), k] \quad (3.21a)$$

$$\boldsymbol{\lambda}(k) = H_{\mathbf{x}(k)} = \phi_{\mathbf{x}(k)} + f_{\mathbf{x}(k)}^T \boldsymbol{\lambda}(k+1) + h_{\mathbf{x}(k)}^T \boldsymbol{\mu}(k) \quad (3.21b)$$

$$H_{\mathbf{u}(k)} = \phi_{\mathbf{u}(k)} + f_{\mathbf{u}(k)}^T \boldsymbol{\lambda}(k+1) + h_{\mathbf{u}(k)}^T \boldsymbol{\mu}(k) = 0 \quad (3.21c)$$

Equations (3.21a) and (3.21b) are the state and co-state difference equations, respectively, while (3.21c) specifies the optimal control variables. Furthermore, the following boundary conditions must be satisfied

$$\begin{aligned} \mathbf{x}(0) &= \mathbf{x}_0 \\ \boldsymbol{\lambda}(K) &= \mathbf{0}. \end{aligned} \quad (3.22)$$

Although expressing a dynamical process, the above minimization problem is, from a mathematical point of view, a Nonlinear Programming Problem (NLP) due to the discrete-time nature of the involved process model, see (Papageorgiou et al., 2015, 2016). However, thanks to the structure of the state equation (3.17), which allows for the state variables to be efficiently eliminated as functions of the control variables, the optimisation problem may be solved, by use of reduced gradients, in the reduced space of the control variables much more efficiently, compared to a general NLP problem with full dimension.

3.4 Numerical Solution and Model Predictive Control

3.4.1 Feasible Direction Algorithm (FDA)

The solution of the formulated OCP is computed by use of the very efficient feasible direction algorithm (FDA) (Papageorgiou et al., 2016; Typaldos et al., 2020b), which exploits the structure of the state equations to map the OCP into an NLP problem in the reduced space of control variables. Thus, the algorithm attempts the calculation of a control trajectory $\mathbf{u}(k), k = 0, \dots, K-1$, which corresponds to a local minimum of the cost function, in the mK -dimensional space, where m is the number of control variables. This marks a substantial

reduction of the problem dimension, as the state variables are eliminated. More specifically, FDA exploits the fact that $g(k) = [\frac{\partial f}{\partial \mathbf{u}(k)}]^T \boldsymbol{\lambda}(k+1) + \frac{\partial \phi}{\partial \mathbf{u}(k)}$ equals the reduced gradient in the mK -dimensional reduced space of the control, if the states and co-states involved in the partial derivatives satisfy the state and co-state equations. The details about this fact may be read in optimal control textbooks (e.g. (Papageorgiou et al., 2015)).

The algorithm is iterative, starting with an initial-guess feasible control trajectory; feasible meaning that it satisfies all state equations and control inequality constraints. The multipliers of the inequality constraints that define bounds are calculated, using (3.21c), as $\boldsymbol{\mu}(k) = -\mathbf{g}(k)/[\partial \mathbf{h}/\partial \mathbf{u}(k)]$ for active constraints and $\boldsymbol{\mu}(k) = 0$ for inactive constraints. At each iteration, using reduced gradient information, a descent search direction in the (reduced) mK -dimensional control space is calculated based on conjugate gradients (or quasi-Newton). Subsequently, a line-search procedure delivers the optimal step along the search direction, and this optimal step leads to an enhanced feasible control trajectory, with improved objective function value. This enhanced trajectory is fed to the next iteration; and so forth, until a sufficiently low reduced-gradient norm is obtained, which marks convergence to a virtually optimal control trajectory. The algorithm guarantees improved objective values at each iteration and features global convergence, from any starting control trajectory, to a local minimum.

The algorithmic steps of FDA are presented below (superscripts (i) indicate the iteration index);

- 1: Receive initial state values.
- 2: Guess an initial control trajectory $\mathbf{u}^{(0)}(k), k = 0, \dots, K-1$
- 3: Calculate states $\mathbf{x}^{(0)}(k)$
- 4: In a unique loop: calculate $\mathbf{g}^{(0)}(k), \boldsymbol{\mu}^{(0)}(k)$ and co-states $\boldsymbol{\lambda}^{(0)}(k)$ for $k = K-1, \dots, 0$, starting with $\boldsymbol{\lambda}^{(0)}(K)$.
- 5: Set iteration index $i = 0$.
- 6: **while** $i < \text{max_iterations}$ **do**
- 7: Specify a descent direction $\mathbf{s}^{(i)}(k), k = 0, \dots, K-1$.
- 8: Specify an optimal scalar step $\xi^{(i)} > 0$ through line optimization.
- 9: Calculate $\mathbf{u}^{(i+1)}(k) = \mathbf{u}^{(i)}(k) + \xi^{(i)} \mathbf{s}^{(i)}(k), k = 0, \dots, K-1$ and $\mathbf{x}^{(i+1)}(k)$ for $k = 0, \dots, K-1$ (apply bounds on control)
- 10: In a unique loop: calculate $\mathbf{g}^{(i+1)}(k), \boldsymbol{\mu}^{(i+1)}(k)$ and co-states $\boldsymbol{\lambda}^{(i+1)}(k)$ for $k = K-1, \dots, 0$, starting with $\boldsymbol{\lambda}^{(i+1)}(K)$.
- 11: Calculate projected gradient when the control bounds are applied.
- 12: **if** not converged **then**
- 13: index increment, $i := i + 1$

```

14:     continue
15: else
16:     break
17: end if
18: end while
19: Generate optimal control input  $\mathbf{u}(k), k = 0, \dots, K - 1$ 

```

3.4.2 Dynamic Programming (DP)

As the OCP at hand is non-convex, FDA may converge to a local minimum. Note that, each single-lane vehicle path can give raise to a local minimum, even though lane changing could possibly improve the cost function. More precisely, gradient-based algorithms search, in each iteration, for descent directions for the control variables. However, in the specific problem, due to the penalisation of lateral acceleration and the rectangular shape of the ellipse in the collision avoidance term (3.14), a controlled path on a lane may feature positive gradients w.r.t. the lateral control, i.e. any lateral deviation from the current path would locally increase the objective value. Thus, if there is no lateral movement in the initial control trajectory, the controlled vehicle will remain in the same lane for the whole time horizon (local minimum), e.g. trapped behind an obstacle on the same lane.

In order to overcome this difficulty and to enhance the quality of the optimal paths delivered by the gradient-based FDA, Dynamic Programming (DP) is first used appropriately to deliver an initial guess trajectory. Despite the involvement of DP in an initial problem solution phase, the overall required computation time for the optimal path generation remains extremely low and certainly real-time feasible. The DP methodology is known to deliver a globally optimal solution trajectory for optimal control problems. However, DP is characterized by computational cost, which increases exponentially with the problem dimensions; and this cost is indeed too high in the present application for efficient real-time path-planning. However, DP is not considered here stand-alone, as it is not real-time feasible for the original path-planning problem. Instead, a simplified problem is solved through DP in very short time, in order to be used as a good initial guess for FDA. Using this combination of methods, the weaknesses of both, i.e. FDA leading to bad local minima and DP being infeasible in real time due to the curse of dimensionality, are strongly mitigated, resulting in good vehicle paths within very short computation times.

In particular, such rough, but globally optimal solutions of the DP simplified problem were found to include lane changes in driving situations where a lane change is indeed beneficial. Note again that the DP algorithm is only used to deliver the initial trajectory for the FDA. Therefore, sacrificing some accuracy due to the simplifications is not important, as the FDA has the final responsibility to deliver an optimal path that satisfies all vehicle tasks and constraints.

The simplifications introduced in the DP problem are:

- A larger step size T (1 s) is used, versus the four times smaller time step (0.25 s) used in FDA and deemed appropriate for path-planning of AVs. As a consequence, the number K of time steps within each planning horizon decreases accordingly.
- The state equation (3.5) is dropped, and the longitudinal acceleration a_x is used as a control variable instead of the jerk.
- The lateral vehicle motion is also simplified by dropping state equation (3.4) and modifying equation (3.2) assuming that the vehicle's lateral position is discrete and lane-based. Specifically, the lateral control is limited to three distinct values at each time step, namely $\{-1, 0, 1\}$, meaning that the vehicle can only apply a lane change towards the adjacent left or right lanes or stay on the same lane.
- Only one lane change is allowed at each planning horizon, in order to reduce the amount of options to be explored by DP.
- The longitudinal acceleration is also roughly discretised and may obtain one out of three values, namely $\{-3, 0, 3\}$ m/s².

Given the above modifications and the fact that state constraints may be directly handled by the DP algorithm, the cost function used with DP is simpler than the corresponding cost function of the OCP and is described as follows

$$J_{DP} = \sum_{k=0}^{K-1} [w_1 a_x^2(k) + w_2 u_y^2(k) + w_3 (v_x(k) - v_d)^2] \quad (3.23)$$

where a_x and u_y are the control variables, corresponding to the longitudinal acceleration and the discrete lateral movement, which reflects the lane changes. Collision avoidance, road departure avoidance and suppression of negative speeds are taken care by the DP algorithm directly, as will be discussed later.

In more detail, the vehicle kinematics (with bigger time step T) in the simplified DP problem are considered as follows

$$x(k+1) = x(k) + v_x(k)T + \frac{1}{2}a_x(k)T^2 \quad (3.24)$$

$$y(k+1) = y(k) + u_y(k) \quad (3.25)$$

$$v_x(k+1) = v_x(k) + a_x(k)T \quad (3.26)$$

where $x(k), y(k), v_x(k)$ correspond to the state variables of the simplified problem and reflect the vehicle's longitudinal position, lane and speed at discrete times $k = 0, \dots, K - 1$, respectively. The control variables a_x, u_y reflect the longitudinal acceleration and lateral change of lane, as mentioned. The state and control variables are bounded within the following feasible regions

$$\mathbf{x}(k) \in \mathbf{X} \quad (3.27)$$

$$\mathbf{u}(k) \in \mathbf{U} \quad (3.28)$$

where the admissible state region \mathbf{X} does not allow for road departures, vehicle interference or negative longitudinal speeds; while the admissible control region \mathbf{U} contains the mentioned discrete values of the control variables in the simplified problem, i.e. $a_x \in \{-3, 0, 3\}$ m/s² and $u_y \in \{-1, 0, 1\}$.

After discretisation of the longitudinal position, speed and acceleration with consistent respective increments, the standard discrete DP algorithm is designed as follows. The algorithm starts with time step $K - 1$ and advances backward, step-by-step. At every step k , all discrete states are branched into all possible transitions (reflecting all combinations of the discrete a_x and u_y values), whereby infeasible transitions (road departure, obstacle collision, negative longitudinal speed) are ignored. For each feasible discrete state, the corresponding optimal controls, along with the corresponding optimal cost-to-go value, are stored. The algorithm ends, when the initial state at $k = 0$ has been evaluated. Eventually, starting from the given initial state and progressing forward step-by-step by following the respective optimal controls at each encountered discrete state, the optimal trajectories are obtained.

In view of the positive semi-definite nature of the objective function (3.23), alternative forward-branching procedures can be employed for the solution of the simplified DP problem, such as branch-and-bound or DFS (Depth-First-Search)-based approaches (Cormen et al., 2001). In the current work, in addition to the aforementioned standard DP, a forward-branching procedure, which employs DFS-based branching (Bar-Yehuda et al., 1989; Cho, Shaw, 1997), was also tested. In this approach, states $\mathbf{x}(k)$ are considered as nodes that can be branched, by application of discrete control combinations, to corresponding subsequent nodes $\mathbf{x}(k + 1)$, whereby transitions that lead to infeasible states (nodes) are ignored. The initial state (root) $\mathbf{x}(0)$ is first branched towards one single (feasible) transition (i.e. using a specific control combination) all the way, until a state (node) $\mathbf{x}(K)$ of the final time K is reached. The cost of this path (trajectory) is provisionally stored as the current optimal cost. Then, the algorithm backtracks continuously, according to the DFS procedure, and creates new paths to the final time K if the arrival cost of all intermediate nodes is lower than the provisional optimal cost. In case a new complete path from $k = 0$ to K is found, which has lower cost, the previous optimal cost is replaced. The algorithm ends when all open (not branched) nodes have a higher arrival cost than the current optimal cost. Thus, the algorithm

may leave many feasible nodes unvisited, which leads to corresponding reduction of the computational effort, without affecting the optimality of the solution. Remarkably, this DFS-based procedure, which leads to identical solutions as the standard DP algorithm, was found to take a computation time ten times lower, on average, compared to the standard DP algorithm for the problem at hand. Specific run times for all employed algorithms are reported in the next section.

More formally, the algorithmic steps of the forward-branching procedure are as follows:

- 1: Initialize: root vertex p (initial state), $J_{Best} = \infty, k = 0$
- 2: Call DFS algorithm for traversing, recursively:
- 3: **procedure** DFS(p)
- 4: label p as visited
- 5: **if** $k > K$ or $p ==$ infeasible **then**
- 6: **return**
- 7: **end if**
- 8: **if** $k == K$ and $J_{DP,p} \leq J_{Best}$ **then**
- 9: $J_{Best} = J_{DP,p}$
- 10: **else if** $J_{DP,p} \geq J_{Best}$ **then**
- 11: **return**
- 12: **end if**
- 13: **for** each child q (combination of $\mathbf{u} \in \mathbf{U}$) of p **do**
- 14: **if** q is undiscovered **then**
- 15: calculate $x(k+1), y(k+1), v(k+1)$ and $J_{DP,q}$
- 16: **if** $\mathbf{x} \notin \mathbf{X}$ **then**
- 17: flag q as infeasible
- 18: **end if**
- 19: $k = k + 1$
- 20: call DFS(q)
- 21: **end if**
- 22: **end for**
- 23: **end procedure**

where p is the parent nodes, q are the children nodes of a parent p , $J_{DP,q}$ is the cost obtained at a node q and J_{Best} is the optimal cost.

The complexity of both DP algorithms can be considered to be known beforehand and constant, due to the fact that the dimensions of the problem are fixed. Specifically, the problem is solved for fixed time-horizon, which means that the dimensions of the control and state variables are not changing. Of course, the actual execution complexity is reduced due to a number of factors, such as: surrounding vehicle density (affecting the amount of feasible

states), current lane (one or both lateral movements possible), the size of a time step, the time-horizon length.

Due to the introduced simplifications, the resulting DP optimal control trajectories have to be processed appropriately, before being used as an initial control trajectory for FDA, so as to be consistent with FDA control inputs. More specifically, the longitudinal acceleration resulting from the DP procedure is transformed to the corresponding jerk via simple Euler differentiation; while any lane changing delivered by the DP procedure is transformed into a lateral acceleration trajectory that leads to the intended lane change also in the FDA settings.

3.4.3 Safety Override

The proposed optimal control approach, with the use of a penalty function (3.14) for collision avoidance does not absolutely guarantee crash-free path generation. Although in the vast majority of cases, the solution of the optimization problem produces trajectories that do not include crashes, thanks to the selected high value of the corresponding weighting parameter, there are very rare cases where, due to the conflicting goals of the collision avoidance term versus the desired speed term, a trajectory including a crash may result. These two terms may be conflicting, as the first term may be striving to decrease the ego vehicle speed, in presence of a slower leading obstacle, while the second term is striving to increase the speed towards the desired speed. A balance between these two terms is typically reached (through the corresponding penalty weights in (3.12)), which guarantees efficient vehicle advancement while suppressing collisions. However, under extreme conditions, e.g. in high density scenarios, the possibility of a collision cannot be utterly excluded.

In order to avoid such decisions and ensure safety, an emergency rule is activated if the solution procedure produces an unsafe path. In this case, the just generated path is dropped, and a new one is generated, with reduced desired speed and planning horizon. Specifically, the desired speed is set equal to a percentage of the leading obstacle's speed (e.g. 95% of obstacle's current speed) and the planning horizon is reduced by half. Both these measures reduce the size of the desired-speed term and enable crash-free vehicle advancement. In particular, the reduction of the planning horizon is helpful because the objective function (3.12) is additive over the time steps; thus, in some rare driving scenarios, it may appear less costly to crash with the leading vehicle for the first few time steps and then achieve all goals for the rest of the planning horizon; instead of avoiding the collision and bearing a high cost each time step due to the desired-speed deviation. Factually, these actions definitely suppress collisions, as the desired speed term is not competing with the collision avoidance term anymore.

3.4.4 Model Predictive Control

Summarising, the presented numerical solution approach requires, as input data, the current (initial) ego vehicle (EV) state, as well as the current and future positions and speeds of obstacle vehicles (OVs); to produce optimal EV controls and states over a future time horizon KT . This is an open-loop solution, and, given the dynamic environment (moving OVs), the time horizon should be long enough to anticipate and prepare for future situations and avoid myopic control actions. On the other hand, as time advances, the uncertainties related to the changing environment (actual versus predicted OV movement) and to the actual vehicle advancement (versus the open-loop solution) increase, as increasing deviations from the assumed predictions are inevitable. To address these uncertainties, the open-loop solution procedure is cast in a model predictive control (MPC) frame, whereby the solution is re-computed online, using the same horizon KT (rolling or receding horizon), whenever substantial changes regarding the initial predictions are detected at any time before the end of the time horizon. The new computation uses updated initial states and updated predictions about the movement of OVs. This calls for computation times smaller than the path update period, something that is indeed satisfied by the presented efficient solution procedure.

Considering traffic flow with many vehicles, as considered in the subsequent simulation investigations, three types of vehicles are distinguished:

1. Manually driven vehicles, which, in the simulation investigations, are navigated by the employed microscopic simulator (Aimsun).
2. Automated vehicles without V2V communication capabilities, which are navigated according to the presented procedure. Such vehicles rely only on their own sensors to sense the current position and speed of other surrounding vehicles of any type, whose paths are predicted simply by assuming that they will keep their current lane and speed fixed over the EV planning horizon KT .
3. Connected automated vehicles (with V2V communication capabilities), which are also navigated according to the presented procedure. However, such vehicles broadcast their latest path decision to other surrounding connected vehicles; and can receive the latest path decisions of surrounding connected vehicles. Note that this information is broadcasted asynchronously, i.e. a path-planning decision by a vehicle is broadcasted as soon as it is produced. Thus, connected AVs rely also on their own sensors to sense the current position and speed of other surrounding vehicles; but, in addition, they receive the latest path planning decision by other surrounding vehicles of the same type.

In the scenarios tested in this work, two cases of AV connectivity are considered:

- Non-connected automated vehicles
- Connected automated vehicles.

Both cases are tested for different penetration rates, along with manually driven vehicles, which are guided by the microsimulation platform.

As evidenced by the above statements, for manually driven vehicles and non-connected AVs, no enhanced information is available, i.e. the non-connected AVs have information about the surrounding vehicles only through their sensors. Consequently, they only know the initial position and speed of the obstacles. So, in order to predict their surrounding vehicles path, the AVs assume that all the obstacles keep their speed and lane fixed over the planning horizon.

On the other hand, connected AVs, having some obstacles of the same type, enjoy enhanced information exchange with such obstacles. Specifically, connected obstacle vehicles (which are also controlled by the proposed approach) send their last generated path to other connected AVs around them. Thus, whenever a connected AV needs to generate a path, it requests the last generated paths of its surrounding connected obstacles, which contains not only the initial position and speed, but also the acceleration and the lane information over the planning horizon. Note that the communication between the connected vehicles is asynchronous, which means that each controlled vehicle generates its path independently, in terms of time, of the other vehicles. Thus, the time a surrounding connected OV last generated its path may deviate from the time that the EV is calculating its own path, e.g., the EV is calculating at time k , while the OV had calculated its path at time $k - n$. For this reason, the available decided path for each connected OV extends up to time $k + K - n$, where K is the time horizon, and must be extended for the $n - 1$ missing time steps. This is done by considering that the OV speed and lane, for these remaining time-steps, remain constant and equal to the last communicated values (of time $k - n + K$).

As mentioned earlier, the EV path is re-generated in real time to address evolving deviations from the last predicted driving conditions. More specifically, a new updated path is generated in the following cases:

- The EV has driven for the duration of half planning horizon ($KT/2$) according to the last generated trajectory. The plan is then updated, even if no deviations are observed, because application of the second half of the last path may lead to myopic actions.
- One or more surrounding vehicles deviate substantially from their predicted paths, e.g. an OV changed lane or changed its speed significantly, compared to its predicted movement.
- A new vehicle enters into the planning zone around the EV, corresponding to a new OV that was not accounted for in the last EV planning.

- The controlled vehicle cannot track the produced path (e.g. when certain safety-related lane-changing restrictions are violated in the microscopic simulation environment).

It may be useful to highlight that the pursued OCP approach aims at minimising an objective function, which reflects the various vehicle tasks, the successful fulfillment of which may entail complex vehicle manoeuvres. In other words, the goal of the vehicle movement strategy is not to track a pre-specified vehicle path, but to create an opportune path; therefore, classic stability is not a major issue here, the main interest focussing on the quality (in terms of the degree of task fulfillment) of the created path.

3.5 Simulation Testing and Results

3.5.1 Simulation Environment

In order to evaluate the proposed path-planning approach, including its MPC-based application, in realistic environments, the procedure was implemented in the Aimsun's (Aimsun Next, 2019) micro-simulation platform with the use of both provided API and SDK tools to integrate the developed path-planning procedures in the traffic context. This implementation enables the investigation of how vehicles, guided by our path-planning approach, interact with each other and with manually driven vehicles, that emulate human driving, in countless driving situations occurring for a variety of traffic conditions. Due to the discrete nature of the micro-simulator with respect to lateral vehicle movement, the produced path of each AV must be modified appropriately to enable its application within the simulator. Specifically, Aimsun does not allow for continuous lateral vehicle positioning or movement, other than discrete lane assignment and instantaneous (vertical) lane changing. Therefore, the continuous lateral AV movements, produced by the path-planning approach, must be translated appropriately in terms of Aimsun lane positioning. For example, in a three-lane motorway section with each lane being 3 m wide, Aimsun assumes that the right-most lane is lane 1, the middle lane is lane 2 and the left-most lane is lane 3. Thus, for Aimsun usage, the lateral AV position at any time must belong to one of these lanes. On the other hand, in the proposed path-planning approach, lateral vehicle position is continuous (as in real conditions), hence, the AV assignment to a discrete lane in the simulator is effectuated according to the AV lateral position: if the AV lateral position is within a range $[0, 3]$ m, the AV is assigned to the right-most lane; if it is within range $[3, 6]$ m, it is assigned to the middle lane; and for range $[6, 9]$ m, the AV is assigned to the left-most lane. The same applies also in case the AV is executing a lane change, leading to a "vertical" lane change, as required by Aimsun, according to Fig. 3.3. This issue is not part of the proposed vehicle movement strategy, which, in a real application, would produce a continuous path. It is only for the needs of the commercial micro-simulation platform (Aimsun) that this discretisation is introduced, so

the procedure is deemed to be of minor importance for the designed movement strategy, but is necessary to enable usage of Aimsun for the comprehensive traffic-level evaluation.

In fact, whether a controlled vehicle will apply a lane change or not, depends on the time the OCP decided this lane change to happen. As already mentioned in the previous section, the OCP is solved for a fixed time horizon, K , but only half of this horizon is applied (at most). Thus, if a lane change was decided during the first $K/2$ period, then the vehicle actually applies the lane change manoeuvre; otherwise, the vehicle just stays at its current lane, with its lateral position being the middle of this lane. Note that, in a lane-based highway environment, a vehicle mostly follows a leader or drives on its lane with its desired speed, as the lane-change manoeuvres are not frequent; therefore explicit consideration of wheel steering is not important. On the other hand, when a lane-change manoeuvre is decided, the vehicle must safely drive from the centre of a lane to the centre of an adjacent lane. This lane change manoeuvre, in real application, can be carried out by following pre-fixed lane-change. For example, the following procedure can be followed in real conditions:

- The algorithm decides about when and where to effectuate a lane-change within a time horizon.
- Whenever a lane change is decided, a pre-fixed lane-change trajectory can be applied, e.g. simply using a sigmoid function from the centre of the vehicle's lane to the centre of the target lane (Claussmann et al., 2015).

For the simulation investigations, two cases were considered, each of them at different penetrations of AVs:

- No connectivity: Each AV is aware only of the current position and speed of obstacles (via its own sensors).
- Connected automated vehicles: Each AV is aware of the current position and speed of obstacles; in addition, it receives the path-planning decisions of other AVs, which facilitates more accurate short-term prediction of their movement.

In both cases, manually driven vehicles are moved according to Aimsun's Gipps (lane-changing) (Gipps, 1986) and IDM (car-following) (Treiber, Kesting, 2013) models.

All investigations use as a testbed a homogeneous motorway section of 3 km in length, with three lanes (each lane being 3 m wide). Two different levels of inflow, 3.000 veh/h and 5.000 veh/h, into this section are simulated, and vehicle trajectories and traffic conditions are monitored over a simulation horizon of 60 min. Entering vehicles are randomly assigned their characteristics (type of vehicle, dimensions, desired speed, initial lane and time gap). In

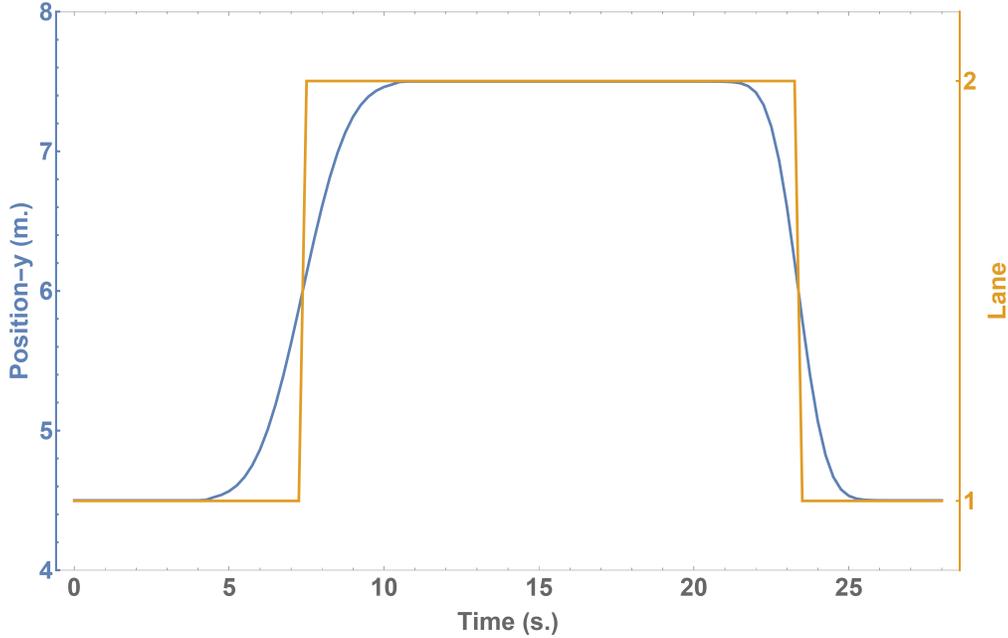


Figure 3.3: AV lateral position trajectory (blue line) and its mapping into Aimsun lanes (orange lines).

particular, vehicle type is selected randomly, according to the examined penetration rate of AVs. All vehicles are "passenger cars" with dimensions selected randomly, from a default range, by the Aimsun simulator. The desired speed of each vehicle (of both types) is selected randomly, with uniform distribution, from a range [80, 120] km/h. The constant time gap, ω , for both automated and manually driven vehicles, is also selected randomly, with uniform distribution, from a range [0.8, 1.8] s (Spiliopoulou et al., 2018).

For AVs, a planning horizon of 8 s is used, with a path-planning step of $T = 0.25$ s. Thus, each plan consists of controls $j_x(k), a_y(k)$ for $K = 32$ time steps. However, as mentioned earlier, half planning horizon is applied at most, which corresponds to 4 s, before a re-plan. The control bounds have been set to $j_x \in [-4.0, 4.0]$ and $a_y \in [-1.5, 1.5]$ for longitudinal and lateral controls, respectively, but it should be noted that high (absolute) control values are virtually never reached. In (3.13) $d = 1.4$; and in (3.14) $p_1 = p_2 = 18$, which lead to rectangular-like ellipses.

In an initial offline trial-and-error procedure, where many different driving situations were

tested, the penalty weights in (3.12) were set to $[w_1, w_2, w_3, w_4, w_5, w_6, w_7] = [1.5, 1.0, 1.5, 0.05, 1.0, 15.0, 15.0]$. The choice of the penalty weights does not depend on the vehicle class or the infrastructure. Problems with multiple sub-objectives, reflected in corresponding weighted penalty terms, have a long history in optimisation. The appropriate specification of weights, so as to reflect the desired relative importance of the sub-objectives, is not a trivial task and is usually addressed via trial-and-error, i.e. by running experiments and evaluating their outcome. This typical trial-and-error procedure was pursued also in this work. Starting from an initial estimate about the appropriate weight values, the following steps were executed:

- (a) Run a simulation involving many different driving circumstances for the AVs.
- (b) Evaluate the AV behaviour during the simulation (e.g. crashes, road departures, travel time etc.).
- (c) If the vehicle behaviour is not satisfactory with respect to some sub-objectives, modify the weights accordingly and go to a).
- (d) Else appropriate weights have been found.

3.5.2 Results

Fig. 3.4 displays the trajectories of several representative AVs, extracted from a simulation scenario with low inflow (3,000 veh/h) and 50% penetration of connected AVs. The trajectories refer to the longitudinal and lateral acceleration and speed; the longitudinal jerk; and the lateral continuous position, produced by FDA, along with the mapping based on lanes, required by the Aimsun microsimulation environment. In Fig. 3.4a, it is observed that the AV cannot reach its desired speed (orange lines). This happens because there is no space to overtake and, consequently, it simply follows the leader by adapting its longitudinal speed in a car-following mode. In Fig. 3.4b, the AV starts with a longitudinal speed equal to its desired speed. In this case, it can be seen that the vehicle needed to slow down and apply lane changes twice, in order to maintain its desired speed. Finally, Figs. 3.4c, 3.4d, combine both first two simpler cases, as the corresponding AVs are manoeuvring through the surrounding slower traffic, managing to achieve their desired speeds for small parts of their trip, while for the rest part they simply follow their leaders, waiting to find appropriate space to overtake, as human drivers would also do. As far as the longitudinal jerk and the longitudinal and lateral accelerations are concerned, the magnitudes are limited and quite smooth, which is good for passenger comfort and fuel consumption.

Figs. 3.5 and 3.6 present results obtained for the two different demand levels, namely for 3.000 veh/h and 5.000 veh/h, respectively. Each figure contains results of different penetration

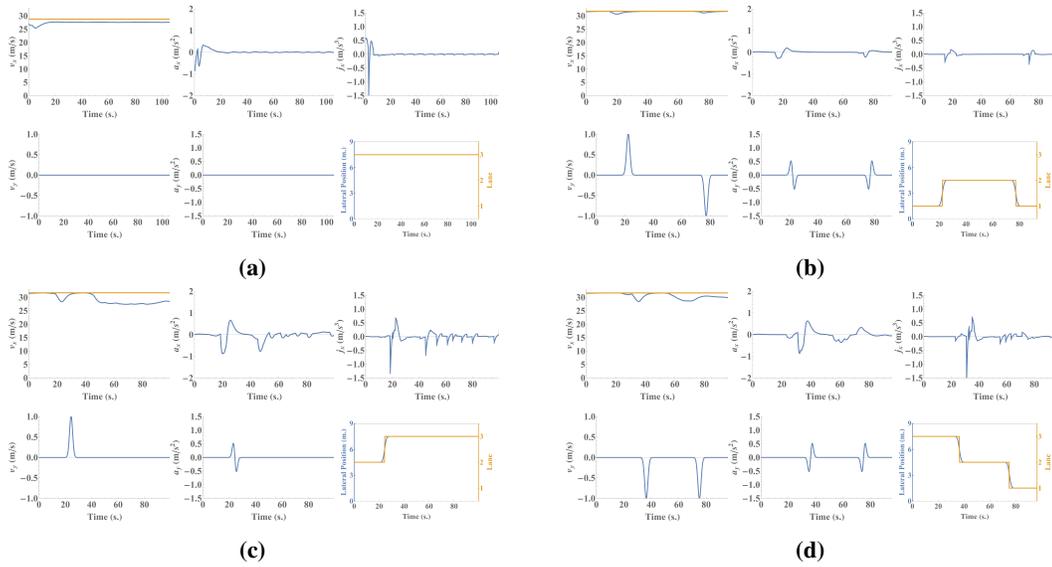


Figure 3.4: Trajectories of several representative AVs during the simulation, displaying longitudinal and lateral speeds (orange line is the longitudinal desired speed) and accelerations, longitudinal jerk, and lateral continuous position produced by FDA (blue lines), along with the corresponding mapping on lanes (orange lines).

rates of AVs. In addition, each figure displays and contrasts results corresponding to the two evaluated cases of AV connectivity: connected AVs (green lines) and non-connected AVs (blue lines). For each case, the solid lines reflect on the average results of the whole vehicle population, including both automated and manually driven vehicles, while the dense-dashed and sparse-dashed lines reflect on the average results of automated and manually driven vehicles, respectively. These summarized results concern the average delay time; the average speed; the average number of lane changes; and the average deviation from the desired speed.

Under both considered demand levels, the online path-planning approach appears to be more efficient in navigating AV speeds closer to the respective desired speeds, compared to manually driven vehicles. Apparently, the suggested approach is more successful at exploiting gaps through traffic and applies "smarter" maneuvers; which leads not only to better performance of each AV, but also to increased overall traffic performance. Specifically, in both Figs. 3.5 and 3.6, as the penetration rate of AVs rises, an increase of the average speed, and consequently a significant decrease of the delay, for the whole traffic, is observed. Note that, this improvement also affects the manually driven vehicles, which appear to also benefit from

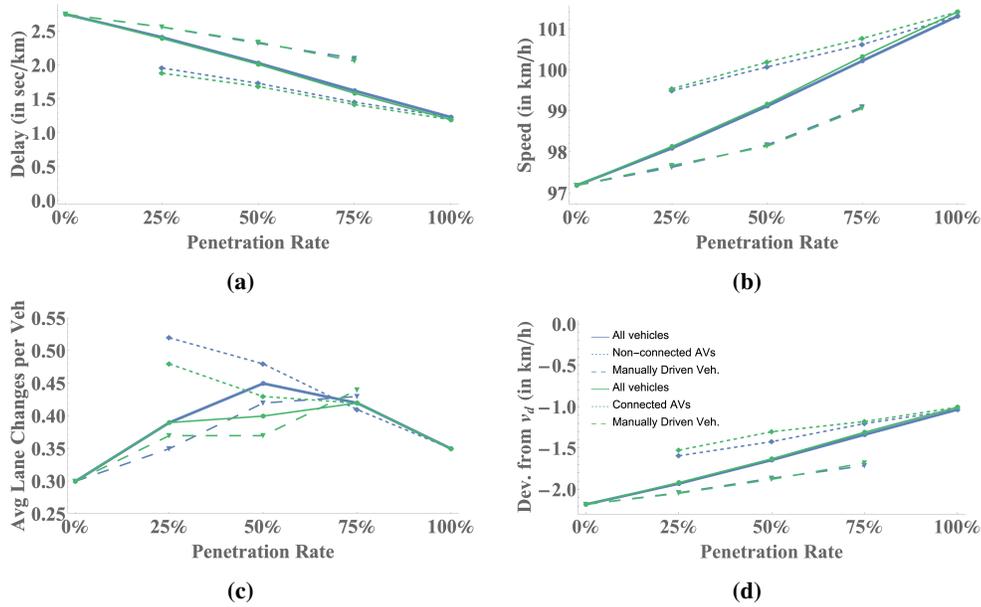


Figure 3.5: Average delay time, speed, deviation from target speed and number of lane changes for: i) connected (green lines); and ii) non-connected (blue lines) AVs; for a demand of 3.000 veh/h. Solid lines represent the average of the whole section, while dense-dashed and sparse-dashed lines reflect on average results for automated and manually driven vehicles, respectively.

the AVs presence and decisions.

In terms of lane changing, the number of lane changes of all vehicles, in the lower demand level (Fig. 3.5c), is higher compared to the higher demand level (Fig. 3.6c). This is due to the density prevailing in each case, where, for the lower demand levels there is more space for the vehicles to apply a lane change in order to overtake slower traffic. Moreover, in Fig. 3.5c, it can be observed that the number of lane changes of AVs is decreasing as their penetration rate rises. This happens as, for low penetration rates, the AVs need to overtake slower traffic, including other slower AVs or manually driven vehicles, which do not have the same capability to achieve their target speed; while for higher penetration rates, the need for overtaking is reduced, as the increased number of AVs ensures speeds closer to the target speed of each vehicle; thus, it is less probable for a vehicle entering the section to face a slower one. On the other hand, in higher demand levels (Fig. 3.6), the number of lane changes of AVs is increasing with their increasing penetration. In this case, where traffic is denser, it is harder, for both the manually driven vehicles and the AVs to overtake. Thus, as the penetration rate of AVs rises, there are more

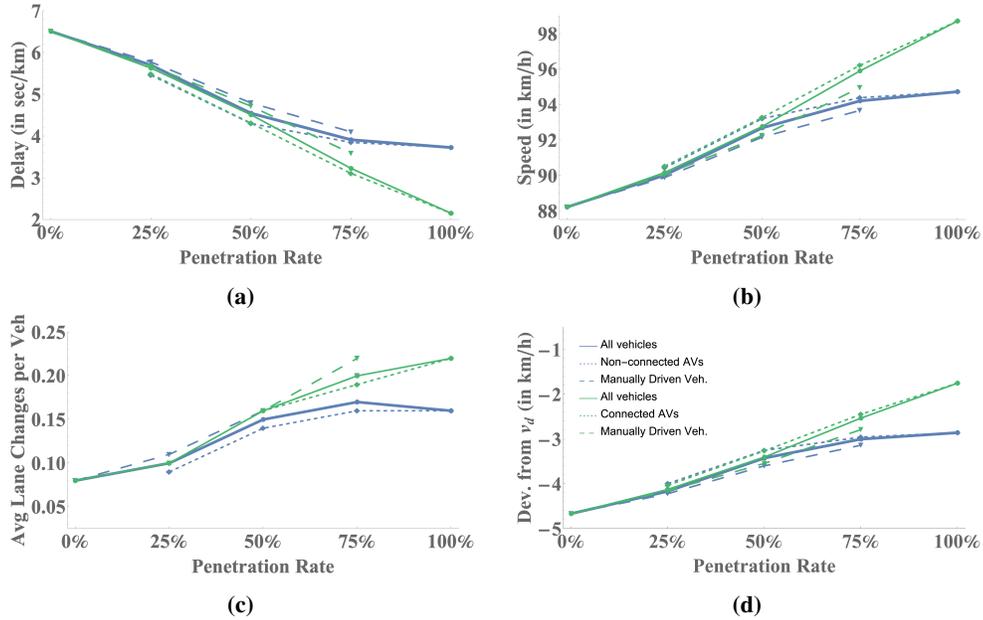


Figure 3.6: Average delay time, speed, deviation from target speed and number of lane changes for: i) connected (green lines); and ii) non-connected (blue lines) AVs; for a demand of 5.000 veh/h. Solid lines represent the average of the whole section, while dense-dashed and sparse-dashed lines reflect on average results for automated and manually driven vehicles, respectively.

overtakes from AVs, which exploit the available space better, helping to maintain increased speed and consequently create spaces for the following traffic. In both demand levels, the AVs efficient maneuvering behavior allows for the manually driven vehicles to increase their lane changes as well.

Contrasting the two types of AVs, the results in Fig. 3.5 indicate only small differences, with connected vehicles being able to achieve slightly better performance in higher penetration rates. This similar outcome is due to the fact that, in lower demand levels, driving space is ample for an AV to maneuver efficiently, which is also evident from Fig. 3.5c, where the difference in the average number of lane changes is moderate. For the same reason, all vehicles (manually driven and automated) do not need to change their speed frequently or strongly, hence the added value of receiving improved information (last path decision) from the surrounding AVs, through the connectivity with other vehicles, is not significant. On the other hand, in Fig. 3.6, where the demand level is higher, vehicle connectivity is seen to have a high impact on the vehicles performance. In this scenario, it is noticed that, although both

types of vehicles manage to achieve improved performance, connected AVs outperform the non-connected ones, as the penetration rate increases, with most noticeable differences in the area of 75%-100%. This is explained as, in denser conditions, the enhanced information that the AVs have about the surrounding traffic enable them to achieve better reactions in need of a lane change. That means, that the connected AVs are able to apply few lane changes due to better timing, compared to the non-connected ones, which also leads to keeping their speed closer to the desired speed.

Finally, Fig. 3.7 reports on the average number of plans (and re-plans) of the two types of AVs, in dependence of the penetration rates and for both demand levels. In Fig. 3.7a, it is noticed that both types of AVs have approximately the same average number of (re-)plans. In this low demand case, all AVs are able to navigate close to their desired speeds, with no significant changes to their predicted paths. The slightly reduced values for the connected AVs are due to their enhanced information, specifically the knowledge of the lane changes that other neighboring AVs have planned to apply. On the other hand, in Fig. 3.7b, where the higher demand level case is presented, it is observed that, as the penetration rate increases, there is an increase of plan numbers for both types of AVs. As also mentioned above for the lane changing behavior, this is because in this case, for low penetration rates, the surroundings of each AV do not change much, hence there are few deviations from the predicted paths for the obstacles. However, as the penetration rate rises, the AVs are more capable in maneuvering through traffic, which increases the need for re-plans. In these conditions, the number of lane changes increases, vehicles may accelerate or decelerate after an overtake or may reach a vehicle downstream, which was not included in their initial prediction. This increase of the average number of plans is different for the two types of AVs, with connected AVs demanding less re-plans compared to the non-connected ones, as their enhanced knowledge allows them to have more accurate view of the surrounding traffic, including the intended lane changes and the tendency of other AVs to accelerate or decelerate.

Regarding computation times, the average CPU-time per planning of an AV path during the simulation is 0.1 s for the DP, 0.01 s for forward-branching DP and 0.01 s for FDA, with the corresponding maximum values during a whole simulation being 0.2 s for DP, 0.06 s for forward-branching DP and 0.2 s for FDA, which indicates that the proposed approach is clearly real-time feasible. In addition, in Table 3.1, the average execution times of the three algorithms, for different planning horizons are reported, which verify their polynomial increase against the time-horizon length; and the real-time applicability of the proposed approach, even for increased time horizons.

Demonstration of the vehicle movements in Aimsun micro-simulation platform is available as videos at <https://bit.ly/3m2nPa2>. Specifically, there are three videos, showing the connected AVs' behavior in both demand levels, i.e. 3.000 veh/h (video-1) and 5.000 veh/h (video-2 and video-3), both for 100% penetration rate. It is evident from the

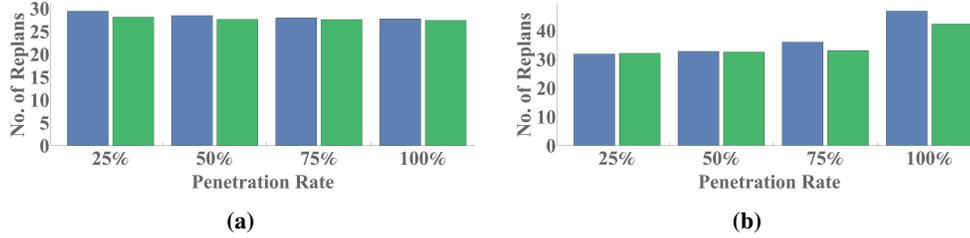


Figure 3.7: Average number of (re-)plans for different penetration rates of the two evaluated cases: non-connected AVs (blue); and connected AVs (green) and for different demand levels: a) 3.000 and b) 5.000 veh/h

videos that the above OCP formulation is conceived for American freeway traffic rules, whereby vehicles may use any lane at any speed and may overtake on the left or right. Adoption of European driving rules, where overtaking is only from left, may be easily accommodated according to (Makantasis, Papageorgiou, 2018).

3.6 Conclusions

Automated vehicle path-planning has been expressed as an optimal control problem. A combination of DP and NLP techniques allows obtaining good local minima for this non-convex optimization problem efficiently. This efficiency facilitates an online MPC-based (re-)planning approach, by observing or receiving information (according to the connectivity features) and predicting the trajectory of the surrounding vehicles and adapting the EV path accordingly. The proposed MPC-based approach is embedded within the Aimsun micro-simulation platform, enabling us to thoroughly examine the behavior of the approach in

Table 3.1: Average execution times of the three proposed algorithms, FDA, DP and Forward DP, for different planning horizons.

Time Horizon (<i>KT</i>)	Average CPU time		
	FDA	DP	Forward DP
6 s	0.002	0.04	0.001
8 s	0.010	0.09	0.010
10 s	0.020	0.12	0.015
12 s	0.035	0.17	0.018

presence of vehicles emulating human driving behavior and in a plethora of realistic driving instances. For the simulation investigations, two cases were considered, each of them at different penetration rates of AVs: i) non-connected vehicles, where each AV is aware only of the current states of obstacles, which are either manually driven or automated; and ii) connected AVs, where each AV is aware of the current states of obstacles, but, in addition, it receives (asynchronously) the path-planning decisions of other AVs.

Demonstration results are reported for a homogeneous motorway stretch and different lane-capacity utilizations. Based on the results, it is concluded that the introduction of AVs, guided by the suggested approach, benefits the overall traffic performance. Specifically, in all scenarios, as the penetration of AVs rises, an increase of the average speed and consequently a decrease of the average delay of both automated and manually driven vehicles are observed. AVs appear to be more effective to navigate closer to their desired speed compared to the manually driven vehicles in all tested scenarios. Specifically, in the case of lower demand level, both non-connected and connected AVs manage to improve their average delay time by 44%, with the speed being increased by 4% compared to the manually driven vehicles. On the other hand, for the higher demand level scenario, the improvement of the average delay time was 38% and 69% for the non-connected and the connected AVs respectively; while the average speed was increased by 6% and 10%, respectively. The improved performance of controlled vehicles stems from their ability to apply smarter manoeuvres, while in very high density or congested cases, the performance of both controlled and non-controlled vehicles is expected to be similar, due to very limited or no space to exploit. Of course, further investigation, in such cases, is required, which is part of future work. As far as the AVs are concerned, in lower demand levels both connected and non-connected AVs perform similarly, due to sufficient space for maneuvering. The superiority of connectivity becomes evident in higher demand levels, as the enhanced information about the surrounding traffic is crucial, and connected AVs appear more efficient at increased penetration rates, due to the improved real-time information that enables more pertinent obstacle movement prediction.

Current and future work is focused on:

- Investigation of scenarios, which feature very high demand levels and congested areas.
- Consideration of platooning and multiple vehicle coordination and study of the synergistic effects and impact on the overall traffic flow.
- Application of a similar approach for urban road networks.
- Introducing on-ramps, off-ramps and emergency cases, e.g. controlled vehicles need to create appropriate space for an ambulance.

Acknowledgments

The research leading to these results has received funding from the European Research Council under the European Union's Horizon 2020 Research and Innovation programme / ERC Grant Agreement n. [833915], project TrafficFluid.

Chapter 4

Vehicle Trajectory Specification in Presence of Traffic Lights with Known or Uncertain Switching Times

4.1 Abstract

The main purpose of this work is to generate optimal trajectories for vehicles crossing a signalized junction, with traffic signals operated in either fixed-time or real-time (adaptive) mode. In the latter case, the next switching time is decided in real time based on the prevailing traffic conditions and is therefore uncertain in advance. The GLOSA (Green Light Optimal Speed Advisory) problem is addressed by using traffic lights information and calculating a trajectory and velocity profile for the vehicle based on the vehicle's initial state (position and speed) and a fixed final destination state. At first, an appropriate optimal control problem is formulated and solved analytically via Pontryagin's Minimum Principle (PMP) for the case of known switching times. Subsequently, for the case of real-time signals, availability of a time-window of possible signal switching times, along with the corresponding probability distribution, is assumed, and the problem is cast in the format of a stochastic optimal control problem and is solved numerically using Stochastic Dynamic Programming (SDP) techniques. Application results, for various driving scenarios, of the deterministic approach, which considers the case of known switching times; and a comprehensive comparison of the stochastic GLOSA approach with a sub-optimal approach are presented. In particular, it is demonstrated that the proposed SDP approach achieves better average performance compared to the sub-optimal approach due to the better (probabilistic) information on the traffic light switching time.

4.2 Introduction

In view of cheap energy resources shortage and excessive environmental pollution, it is necessary for transportation systems to operate with increased fuel efficiency. In the case of road vehicles, fuel efficiency relates to economic aspects, as fuel economy means less expenses for the driver, but also to the protection of the environment in an era of climate crisis. Indeed, a wide range of technologies have been developed in recent years to decrease fuel consumption of vehicles, including efficient engines, adjusted vehicle designs and lighter chassis. Furthermore, considerable efforts in the development and deployment of efficient intelligent transportation systems (including real-time traffic signals) lead to reduced congestion and fuel consumption.

Traffic signals guarantee, in the first place, the safe crossing of vehicles at urban junctions in cities around the world. Clearly, enforcing safety via traffic lights implies that some vehicles will have to stop in front of a red light; and then accelerate after the traffic light switching to green; something that affects the fuel consumption of concerned vehicles. To reduce the resulting vehicle delays and number of stops, several algorithms have been proposed and deployed over the past decades, aiming at optimizing the traffic signals operation, see (Hounsell, McDonald, 2001; Papageorgiou et al., 2003) for overviews. Fixed-time signal plans are derived off-line for respective times-of-day (e.g., morning peak hour, off-peak etc.) by use of appropriate optimization codes, based on historical constant demands; and are applied without deviations; this implies that switching times of the traffic lights are always known in advance. In contrast, real-time (or traffic-responsive or adaptive) signal control strategies make use of real-time measurements to calculate in real time suitable signal settings. Depending on the employed signal control strategy, the control update period may range from one second to one signal cycle; clearly, for real-time signals, the next switching time is not known before the switching decision is actually made. It should be noted that real-time strategies are deemed to be more advanced and potentially more efficient than fixed signal operation. Fuel consumption is increasingly considered as an optimization or evaluation criterion while developing and deploying signal control systems (Jamshidnejad et al., 2017).

Consider a vehicle approaching a red traffic light at a given speed. If the vehicle continues driving at that speed, it may reach the traffic light after it has switched to green, in which case no fuel-intensive acceleration needs to be applied; however, the constant-speed vehicle might also reach the traffic light before the green switch, in which case it will have to stop and accelerate to a higher speed after the green switch. On the other hand, if the vehicle decelerates smoothly in view of the red light, this may prove beneficial or not, depending on the time of the green switch. This dilemma of vehicle movement towards a red traffic signal may be addressed by appropriately designed systems. Some early such systems (Van Leersum,

1985) were displaying on road-side dynamic advisory speed signs the speed that would allow a vehicle to cross the downstream signalized junction at green. With recent and emerging advances in vehicle communications, the current state and timing of a traffic signal can be transmitted to equipped vehicles (or apps therein) to enable sensible approaching speed decisions. Based on signal information, it is possible to guide the driver (or an automated vehicle) all the way from the current state to the traffic signal by giving speed advice, which ensures that, by the time the vehicle reaches the traffic signal, it will be green; or, more specifically, that the fuel consumption and emissions will be minimized. Systems (or apps) optimizing the vehicle approach to traffic lights are often referred to as Green Light Optimal Speed Advisory (GLOSA) systems (Stahlmann et al., 2016).

In the case of fixed signals and prior knowledge of the next switching time, e.g. via broadcasting of corresponding messages by the signal controller, the problem of how to optimize the approach to traffic signals has been addressed in different ways. In (Richter, 2005), speed profiles have been compared to their energy demand; however, only a limited set of profiles has been considered, and, more generally, there is a lack of mathematical justification that the derived speed profile is indeed optimal. Rule-based algorithms have been employed in various works to produce advisory speeds for vehicles approaching traffic signals, so as to reduce fuel consumption and emissions, see e.g. (Katsaros et al., 2011; Ma et al., 2018; Sanchez et al., 2006); clearly, rule-based algorithms may deliver sub-optimal results, in particular when the dynamic vehicle kinematics are not accounted for, i.e. when no vehicle acceleration, but only the vehicle speed, is involved. Optimal control approaches, considering explicitly the vehicle kinematics, are, by their nature, more pertinent in producing fuel-optimal speed profiles for vehicles approaching fixed traffic signals. Such an approach is proposed in (Lawitzky et al., 2013), attempting the minimization of vehicle's physical work; the resulting optimal control problem is solved analytically via Pontryagin's Minimum Principle (PMP). This paper also presents an optimal control approach for fuel minimization with more pertinent final state conditions for the vehicle, consisting of a final position sufficiently downstream of the signals and a high final speed, so as to avoid distorted or non-comparable vehicle trajectories.

The situation becomes more complicated when real-time signals with very short (e.g. second-by-second) control update periods are present, in which case exact prior knowledge of the next switching time is not available even with the signal controller. In this case, the best available knowledge about the next signal switching can be presented as an estimate or as a probabilistic distribution. A system called SignalGuru (Koukoumidis et al., 2011) does not require communication to the infrastructure, but relies solely on a collection of mobile phone data to detect and predict the traffic signal timing. The mobile phones detect current traffic signals with their cameras, collaboratively communicate and learn traffic signal timing patterns, and predict their future timing. Reportedly (Koukoumidis et al., 2011), SignalGuru's

estimates come on average within 0.66 s for fixed traffic signals; and within 2.45 s for real-time traffic signals; of the actual switching times. Such an estimate of the next switching time may then be used as a proxy for any of the GLOSA approaches that need a known switching time, possibly with appropriate extensions to account for the consequences of estimation inaccuracy.

Richer probabilistic information than a single estimate for the next signal switching time can be produced in form of a probability distribution for the next switching time within a short-term future time-window by use of statistics from previous signal operations over a sufficiently long rolling horizon. Such an approach to produce probability distribution for the next switching time, based on past signal switching, is proposed in (Mahler, Vahidi, 2012); however, the available probability distribution is used heuristically (rather than optimally), to accordingly time-weight the objective function, within a deterministic optimal control problem that is solved via a Dynamic Programming algorithm. An appropriate utilization of the available probabilistic distribution of the next signal switching time within a stochastic optimal control problem is proposed in (Lawitzky et al., 2013), and the problem is solved via a discrete Stochastic Dynamic Programming (SDP) algorithm; however, the formulated optimal control problem extends only up to the point in time where the next switching time becomes known, thus neglecting the cost incurred after this time until the vehicle's final state is reached, something that may lead to non-commensurable vehicle trajectories and skewed solutions.

This paper considers the problem of producing fuel-optimal vehicle trajectories for a vehicle approaching a traffic signal (GLOSA) for both cases of known and probabilistic switching times. For the first case, the problem is formulated as an optimal control problem with pertinent final state for the vehicle and is solved analytically via PMP. Subsequently, the case where the traffic light's switching time is stochastic, with known probability distribution, is addressed, and the problem is cast in the format of a stochastic optimal control problem, which is solved numerically using SDP techniques. In contrast to previous works, the stochastic optimal control problem is designed to extend up to the vehicle's final state by exploiting appropriately the analytic solution of the fixed switching time problem derived earlier, something that guarantees that the obtained solution is consistent, independently of the actual switching time occurrence. The presented approach may be readily generalized to allow for switching time decisions to be taken at any time in advance of the actual switching time, not necessarily only within the last time period prior to the actual switching time.

The remainder of the paper is organized as follows: in Sections 4.3 and 4.4 the optimal control problems with known signal switching time and uncertain signal switching time, respectively, are formulated and solved. Several demonstration and comparison results of the deterministic and the stochastic GLOSA are presented in Section 4.5; while Section 4.6 concludes this work.

4.3 Optimal Control Problem with Known Signal Switching Time

This section describes in detail the proposed GLOSA approach when the signal switching time is known beforehand. An efficient path-planning procedure for a vehicle starting from an initial state to reach a final state within a free time horizon, with the interference of a traffic light at a given position is designed.

4.3.1 Problem Formulation

More specifically, the vehicle starts from an initial state $x_0 = [x_0, v_0]^T$, where x_0 is a given initial position and v_0 a given initial speed of the vehicle; and aims at reaching a fixed final state $x_e = [x_e, v_e]^T$ within a free (but weighted) time horizon t_e , with x_e and v_e being the final position and speed of the vehicle, respectively; with the restriction that the vehicle cannot pass through the traffic signal position x_1 before the known time t_1 , which is the time that the traffic light turns green from red. The implicit assumption here is that a red light is active when the vehicle appears on the link (at time 0), which is the typical case; however, the generalization to cases where the traffic signal is green at the initial time is easy and will be mentioned later. The objective of the vehicle is to appropriately adjust its acceleration (control variable), so as to minimize fuel consumption, while satisfying the initial and final conditions x_0 and x_e , as well as the intermediate (traffic signal) constraint.

Note that fixing the final position x_e to be sufficiently (e.g. some 50 m) downstream of the traffic signal; and the final speed v_e to be sufficiently (e.g. some 12 m/s) high, guarantee that the problem will not be myopic and will select the optimal trajectory among all relevant and commensurable ones. In fact, if we would set $x_e = x_1$, as in some other works (Lawitzky et al., 2013), i.e. if we would fix the final vehicle position at the traffic signal location, then:

- If we also fix a final speed, this would exclude trajectories that may be fuel-optimal in hindsight, but feature a different speed at the time of crossing the traffic signal;
- If we do not fix the final speed, we would have non-comparable trajectory candidates, i.e. some with low speed at the traffic signal (that would call for fuel-consuming acceleration eventually); and others with higher speed (that are lower-cost in hindsight).

It should be noted that an even less myopic policy may result if we consider two, instead of one, downstream traffic lights, which calls for substantial extension of the present approach and is the subject of current investigations.

The minimization problem outlined above is formulated as an optimal control problem, which accounts for the vehicle kinematics via the following state equations:

$$\dot{x} = v \quad (4.1)$$

$$\dot{v} = a \quad (4.2)$$

where a is the vehicle acceleration which assumes the role of a control variable. The objective is to drive the system from the given initial condition $x_0 = [x_0, v_0]^T$ to the fixed final condition $x_e = [x_e, v_e]^T$ within time t_e , while minimizing the criterion

$$J = w \cdot t_e + \frac{1}{2} \int_0^{t_e} a^2 dt. \quad (4.3)$$

In addition, the green-light constraint, $t_S \geq t_1$, must be fulfilled, where t_S is the time at which the vehicle crosses from the signal position x_1 , i.e. $x(t_S) = x_1$. Note that the utilized acceleration cost term a^2 in the cost criterion was demonstrated in (Typaldos et al., 2020b) to be an excellent proxy for deriving fuel-minimizing vehicle trajectories. More specifically, it has been shown that the simple square-of-acceleration term delivers excellent approximation of vehicle trajectories when compared to more sophisticated optimization methods that use a complex fuel-consumption model in the objective function. Note also that the final time t_e is free, but penalized with the parameter w . The reason for leaving the final time t_e free, is because we want to have a flexible problem formulation that applies to a multitude of specific instances of initial positions (far and close to the traffic signal), initial speeds (low or high) and signal switching times; in fact, it would be too rigid to fix for all these instances the same final time t_e . On the other hand, there may be problem instances that lead to excessive final time, if the latter is not accounted for in the cost criterion. The larger the value of w is, the lower the value of the resulting t_e will be. This, consequently, affects the acceleration cost, which, for higher values of w , will also have increased values in general, e.g. because the vehicle will need to reach the final position sooner. Our purpose is to penalize the free final time in order to avoid excessive final times, but without affecting the fuel-consumption goal too much with high values of w . Fig. 4.1 reflects how the free final time is affected according to w for a specific but typical problem instance (that will be specified later). Clearly, the lowest acceleration cost is achieved for $w = 0$; while, increasing w , leads to moderately increasing acceleration cost and decreasing t_e . A value $w = 0.1$ was found to deliver a reasonable trade-off for a variety of examined problem instances.

If necessary, upper and lower bounds may be applied to speed v and acceleration a . Such constraints would limit the speed between zero and a maximum value; and the acceleration between minimum and maximum values.

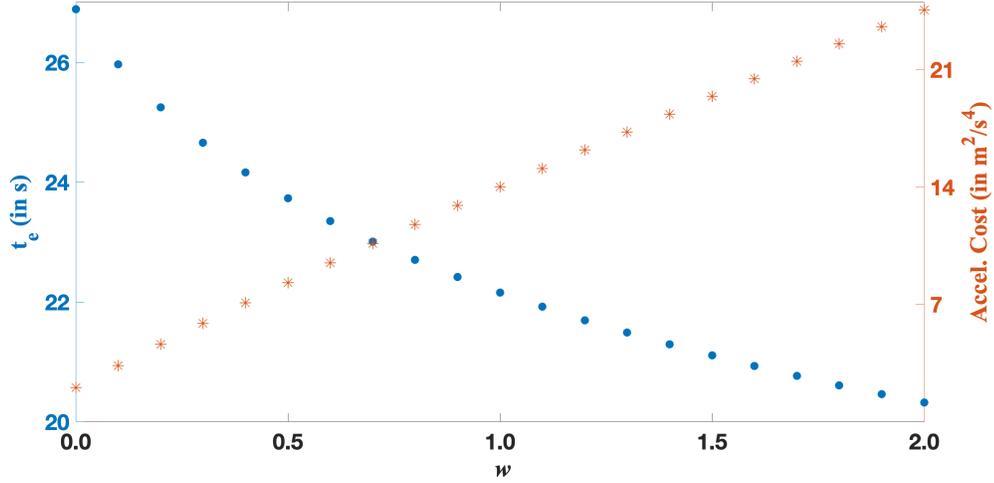


Figure 4.1: Optimal final time t_e (blue dotted marks) and acceleration cost (red star marks) versus the weight w .

4.3.2 Analytical Solution

In order to present the necessary condition and proceed with the analytical solution of the problem, it is customary to use the Hamiltonian function (Papageorgiou et al., 2015; Pontryagin, 2018)

$$H(\mathbf{x}, \boldsymbol{\lambda}) = \phi(\mathbf{x}, \mathbf{a}) + \boldsymbol{\lambda}^T \mathbf{f}(\mathbf{x}, \mathbf{a}) \quad (4.4)$$

where $\boldsymbol{\lambda}$ stands for the co-state vector for the corresponding state equations, and $\phi(\mathbf{x}, \mathbf{a}) = \frac{1}{2}a^2$ is the running cost in the objective function (4.3). For the specific problem, we have the Hamiltonian function

$$H(v, a, \lambda_1, \lambda_2) = \frac{1}{2}a^2 + \lambda_1 v + \lambda_2 a. \quad (4.5)$$

Then we have the following necessary conditions of optimality

$$\dot{x} = \frac{\partial H}{\partial \lambda_1} = v \quad (4.6)$$

$$\dot{v} = \frac{\partial H}{\partial \lambda_2} = a \quad (4.7)$$

$$\dot{\lambda}_1 = \frac{\partial H}{\partial x} = 0 \quad (4.8)$$

$$\dot{\lambda}_2 = \frac{\partial H}{\partial v} = -\lambda_1 \quad (4.9)$$

$$\frac{\partial H}{\partial a} = 0 \quad (4.10)$$

These equations must be satisfied, along with the initial and final state conditions for a specific problem. In addition, due to the free final time t_e , we have an additional boundary condition for optimality (Papageorgiou et al., 2015; Pontryagin, 2018)

$$H(t_e) + w = 0. \quad (4.11)$$

Let us first consider the "unconstrained" problem (UP), i.e. without the intermediate green-light constraint. Equations (4.8)-(4.10) readily yield an analytic linear-in-time optimal acceleration solution for UP (Ntousakis et al., 2016; Rios-Torres, Malikopoulos, 2016b)

$$a(t) = c_1 t + c_2 \quad (4.12)$$

and, after integration according to (4.6), (4.7), also the corresponding speed and position solutions

$$v(t) = \frac{1}{2} c_1 t^2 + c_2 t + c_3 \quad (4.13)$$

$$x(t) = \frac{1}{6} c_1 t^3 + \frac{1}{2} c_2 t^2 + c_3 t + c_4 \quad (4.14)$$

where c_1, \dots, c_4 are four integration constants, which, along with the optimal final time t_e , may be specified as the solution of a system of five algebraic equations, comprising the initial and final states and the final time condition (4.11).

We now proceed with a "constrained" problem (CP), wherein we require the vehicle position at some given time t_c to be at the traffic signal, i.e. $x(t_c) = x_1$. In this case, all necessary conditions (4.6)-(4.11) remain the same, but there is an additional necessary condition that must be satisfied (see (Papageorgiou et al., 2015), Chapter 10.5.2), which, for

the present problem, implies that the co-state $\lambda_1(t)$ may be dis-continuous at the intermediate time t_c . This and (4.8)-(4.10) then yield a continuous two-branch piece-wise linear optimal acceleration solution for CP as follows

$$a(t) = \begin{cases} c_1t + c_2 & 0 \leq t \leq t_c^- \\ c_5t' + c_6 & t_c^+ \leq t \leq t_e \end{cases} \quad (4.15)$$

and, after integration according to (4.6), (4.7), also the corresponding speed and position solutions

$$v(t) = \begin{cases} \frac{1}{2}c_1t^2 + c_2t + c_3 & 0 \leq t \leq t_c^- \\ \frac{1}{2}c_5t'^2 + c_6t' + c_7 & t_c^+ \leq t \leq t_e \end{cases} \quad (4.16)$$

$$x(t) = \begin{cases} \frac{1}{6}c_1t^3 + \frac{1}{2}c_2t^2 + c_3t + c_4 & 0 \leq t \leq t_c^- \\ \frac{1}{6}c_5t'^3 + \frac{1}{2}c_6t'^2 + c_7t' + c_8 & t_c^+ \leq t \leq t_e \end{cases} \quad (4.17)$$

where $t' = t - t_c$ and c_1, \dots, c_8 are eight integration constants, which, along with the optimal final time t_e , may be specified as the solution of a system of nine algebraic equations, comprising: the initial and final states; the final time condition (4.11); the continuity conditions for control and states $a(t_c^-) = a(t_c^+), v(t_c^-) = v(t_c^+), x(t_c^-) = x(t_c^+)$; and the intermediate condition $x(t_c) = x_1$.

We denote $J_{CP}(t_c)$ the optimal objective value of CP as a function of t_c and J_{UP} the optimal objective value of UP. Since CP was obtained by additional constraining of UP, we have $J_{CP}(t_c) \geq J_{UP} \forall t_c$, and, in fact, $J_{CP}(t_c)$ obtains its minimum value for $t_{c(S,UP)}$, where $t_{(S,UP)}$ is the time when the UP position trajectory reaches the traffic signal position x_1 . It was also found that, as expected, the function $J_{CP}(t_c)$ increases monotonically for $t_c \geq t_{(S,UP)}$. These findings suggest the following procedure for finding the solution of the original GLOSA problem in presence of a traffic signal at x_1 and switching time to green t_1 :

1. Solve UP; if $t_{(S,UP)} \geq t_1$, the GLOSA problem is solved, as the UP solution does not violate the green-light constraint; else
2. Solve CP with t_{c_1} to obtain the GLOSA solution.

As mentioned, the solution of UP or CP for a specific problem instance calls for the solution of a corresponding system of five or nine, respectively, algebraic equations. Such solutions were obtained analytically (yielding lengthy formulas) using symbolic differentiation tools (e.g. Mathematica (Wolfram Research, Inc., 2016)). The numerical execution of those analytical

formulas to obtain the solution of UP or CP, and hence of the GLOSA problem for a specific problem instance, takes only fractions of a second of computation time and may therefore take place within the vehicle (or app) once the next switching time becomes known. Note that it may be reasonable to continuously update the vehicle trajectory (in a model predictive control (MPC) loop) to account for possible deviations from the first calculated vehicle trajectory, which may occur due to a variety of disturbances, including a slower vehicle ahead.

For a given junction, the final state is the same for any initial vehicle state x_0 and any switching time t_1 . Hence, the optimal value of criterion (4.3) of the deterministic GLOSA problem depends on these variables, and we denote it $J_{DG}^*(x_0, t_1)$ for later use.

In the above formulation, there is an implicit assumption that a red light is active when the vehicle appears on the link (at time 0), which is the typical case. The approach, however, can be easily generalized to the case where the traffic signal is green at the initial time. In this case, we also need to know the switching time, say T_1 , from green to red. Note that, for similar reasons as above, the function $J_{CP}(t_c)$ increases monotonically also for $t_c \leq t_{S,UP}$, hence the above mentioned procedure is generalized as follows for the case of green initial light:

1. Solve UP; if $t_{(S,UP)} \geq t_1$ or $t_{(S,UP)} \leq T_1$, the GLOSA problem is solved, as the UP solution does not violate the green-light constraint; else
2. Solve CP twice, first with t_{c_1} and second with t_{c_1} ; yielding the respective optimal values J_{CP_1} and J_{CP_1} . The solution with the smaller objective value is the GLOSA solution of the generalized problem.

4.4 Stochastic Optimal Control Problem with Uncertain Signal Switching Time

The traffic light switching time may not be known or be subject to short-term decisions in dependence of the prevailing traffic conditions because of real-time signals. In this case, based on appropriate statistics from past signal switching activity, availability of a time-window of possible signal switching times, along with the corresponding probability distribution, may be assumed, in which case the problem can be cast in the format of a stochastic optimal control problem, which may be solved numerically using SDP techniques. To this end, the analytical solution of the deterministic GLOSA optimal control problem derived in the previous section is used within the stochastic approach, as will be explained in this section.

4.4.1 Problem Variables and State Equations

Since SDP algorithms require a discrete-time system, the discrete-time version of the vehicle kinematics with time-step T is considered, as follows (Bertsekas, 1995)

$$x(k+1) = x(k) + v(k)T + \frac{1}{2}a(k)T^2 \quad (4.18)$$

$$v(k+1) = v(k) + a(k)T \quad (4.19)$$

where $x(k), v(k)$ correspond to the vehicle position and speed at discrete times $k = 0, 1, \dots$ (where $kT = t$, while the control variable $a(k)$ reflects the acceleration that remains constant over each time period k . The state and control variables are bounded within the following feasible regions

$$\mathbf{x}(k) \in \mathbf{X} = [\mathbf{x}_{\min}, \mathbf{x}_{\max}] \quad (4.20)$$

$$a(k) \in U = [a_{\min}, a_{\max}] \quad (4.21)$$

with $\mathbf{x}_{\min}; \mathbf{x}_{\max}$ and $a_{\min}; a_{\max}$ being the lower and upper bounds of the states and acceleration, respectively. The traffic light's switching time k_1 is not known, but it is assumed that a known range $k_{\min} \leq k_1 \leq k_{\max}$ of possible switching times exists, with k_{\min} and k_{\max} being the minimum and maximum possible switching times. Note that real-time signal control strategies must typically respect pre-fixed minimum and maximum bounds for each signal phase, which may be the reason for having the mentioned time-window.

For proper problem formulation, a virtual state $\tilde{x}(k)$ is introduced, that reflects formally the stochasticity of traffic light switching

$$\begin{aligned} \tilde{x}(k+1) &= \tilde{x}(k) \cdot z(k) \\ \tilde{x}(0) &= 1 \end{aligned} \quad (4.22)$$

where $z(k)$ is a discrete stochastic variable defined as

$$z(k) = \begin{cases} 0 & \text{if traffic light switches at time } k+1 \\ 1 & \text{else} \end{cases} \quad (4.23)$$

With (4.22) and (4.23), the virtual state $\tilde{x}(k)$ is either equal to 1, if the traffic light has not yet switched until time $k-1$, or equal to zero if switching occurred at time k or earlier. The virtual state $\tilde{x}(k)$ is assumed measurable, which means that the system knows at each time kT if switching has taken place or not within the last time period $[(k-1)T, kT]$.

The stochastic variable $z(k)$ is independent from its previous values $z(k-1), z(k-2), \dots$

and takes values according to a time-dependent probability distribution $p(z|k)$. Based on the statistics of previous signal switching activity, availability of an a-priori discrete probability distribution $P(k)$, $k_{\min} \leq k_1 \leq k_{\max}$ is assumed, for signal switching within the time-window, where $\sum_{k=k_{\min}}^{k_{\max}} P(k) = 1$. Since no switching takes place for $k \leq k_{\min} - 1$, then

$$p(0|k) = 0 \quad \text{for } k < k_{\min} - 1. \quad (4.24)$$

For $k = k_{\min}$, signal switching may occur with a-priori probability $P(k_{\min})$, therefore

$$p(0|k_{\min} - 1) = P(k_{\min}). \quad (4.25)$$

If the traffic light has not switched at time $k = k_{\min}$, then the probabilities of switching at some later time within the time-window are increased, compared with the respective a-priori distribution, and the updated probabilities may be calculated via "crop-and-scale", meaning that the a-priori probability $P(k_{\min})$ distributed analogously to increase the probabilities of the remaining discrete times within the time-window (Lawitzky et al., 2013). By the same argument, the crop-and-scale procedure for updating the switching probabilities must be executed at each consecutive time step, as long as the switching has not occurred. This update may be done by use of the following crop-and-scale formula that applies for $k_{\min} \leq k \leq k_{\max} - 1$ and for any a-priori distribution $P(k)$

$$p(0|k) = P(k+1) \left[1 + \frac{\sum_{\kappa=k_{\min}}^k P(\kappa)}{\sum_{\kappa=k+1}^{k_{\max}} P(\kappa)} \right] \quad (4.26)$$

where the term in brackets reflects the crop-and-scale update.

4.4.2 Objective Criterion

The cost criterion of the stochastic problem is the same as in the deterministic case (4.3). However, in the stochastic case, the exact value of the criterion depends on the stochastic variable's realization, and therefore its expected value is minimized

$$J = E \left\{ w \cdot t_e + \frac{1}{2} \int_0^{t_e} a^2 dt \right\} \quad (4.27)$$

where the expectation refers to the stochastic variable $z(k)$, $k = 0, \dots, k_{\max} - 1$. Note that, when the switching time becomes known at time k_1 , while the vehicle is at state $x(k_1)$, advancing toward the traffic signal, the problem instantly becomes a deterministic GLOSA problem, and the corresponding optimal cost-to-go is $J_{DG}^*[x(k_1), k_1]$, which will be denoted as the escape cost. After introducing discrete-time notation, it is possible to obtain from (4.27), by the principle of

optimality (Bellman, 1966),

$$J = E\left\{\frac{1}{2} \sum_{k=0}^{k_1-1} a(k)^2 + J_{DG}^*[\mathbf{x}(k_1), k_1]\right\} \quad (4.28)$$

To obtain a formally proper cost criterion, the stochastic variable $z(k)$ and the virtual variable $\tilde{x}(k)$ introduced earlier are used, and the state $x(k_1)$ from the state equations (4.18) and (4.19) is replaced as a function of the state and control of the previous time period (Bertsekas, 1995). This yields the objective function in the required form, as follows

$$J = E\left\{\tilde{x}(k) \sum_{k=0}^{k_{\max}-1} \left[\frac{1}{2}a(k)^2 + [1 - z(k)]J_{DG}^*[\mathbf{x}(k), a(k), k + 1]\right]\right\} \quad (4.29)$$

Equations (4.18)-(4.26), and (4.29) constitute an ordinary stochastic optimal control problem. Denoting the corresponding optimal cost-to-go function by $V[\mathbf{x}(k), \tilde{x}, k]$, the recursive Bellman equation (Bertsekas, 1995) for $0 \leq k \leq k_{\max} - 1$ reads

$$\begin{aligned} V[\mathbf{x}(k), \tilde{x}, k] &= \min_{a(k) \in U} \left\{ E\left\{\frac{1}{2}a(k)^2 + [1 - z(k)]J_{DG}^*[\mathbf{x}(k), a(k), k + 1]\right.\right. \\ &\quad \left.\left. + V[\mathbf{x}(k + 1), \tilde{x}(k)z(k), k + 1]\right\}\right\} \\ &= \min_{a(k) \in U} \left\{ \frac{1}{2}a(k)^2 + p(0|k) \cdot J_{DG}^*[\mathbf{x}(k), a(k), k + 1]\right. \\ &\quad \left. + [1 - p(0|k)] \cdot V[\mathbf{x}(k + 1), \tilde{x}(k)z(k), k + 1]\right\} \end{aligned} \quad (4.30)$$

with boundary condition $V[\mathbf{x}(k_{\max}), 1, k_{\max}]$. Note that the minimum is required with respect to $a(k) \in U$ only, as typical in dynamic programming, which facilitates the numerical solution.

In this formulation, it has been assumed that the decision on traffic light switching is taken within the last time period before the actual switching. The generalization to the case of taking a switching decision κ time periods in advance of the actual switching is easy. To this end, the time k_1 and the definition of the stochastic variable $z(k)$ reflect the decision time (rather than the switching time), and the only change required in the above equations is that the escape function J_{DG} must include as an argument for the switching time $k + \kappa$ instead of $k + 1$.

4.4.3 Numerical Solution Algorithm

For application of the discrete SDP algorithm for numerical solution of the problem, the state and control variables must be discretized. The level of discretization has a significant impact on computational time and memory requirements, but also on the accuracy of the computed

solution. Therefore, an appropriate trade-off should be specified between reasonable computation requirements versus achievable solution quality.

For the discretization, the discrete time interval, T , is set equal to 1 s, which is a reasonable choice for the problem at hand. Then, a general discretization interval Δ for the problem variables is assumed, and the discretization interval of acceleration is set as $\Delta a = \Delta$. From (4.19), it can be noticed that the speed and acceleration intervals are equivalent, and, therefore, the discretization interval of speed can also assume the same value ($\Delta v = \Delta a = \Delta$). Likewise, in view of (4.18), the discretization interval for the position is set as

$$\Delta x = \frac{1}{2} \Delta \cdot T^2 \quad (4.31)$$

Based on these settings, it can be proved that, if $x(k), v(k), a(k)$ are discrete points, then $x(k+1)$ and $v(k+1)$ resulting from (4.18) and (4.19) are also discrete points. To this end, assume that

$$x(k) = n\Delta x \quad (4.32)$$

$$v(k) = m\Delta v = m\Delta \quad (4.33)$$

$$a(k) = l\Delta a = l\Delta \quad (4.34)$$

where n, m, l are positive integers. Using (4.18) gives

$$\begin{aligned} x(k+1) &= n\Delta x + m\Delta \cdot T + \frac{1}{2} l\Delta \cdot T^2 = \\ &= \frac{1}{2} n\Delta \cdot T^2 + m\Delta \cdot T + \frac{1}{2} l\Delta \cdot T^2 = \\ &= \frac{1}{2} \Delta \cdot T^2 \left(n + \frac{2}{T} m + l \right) = \Delta (n + 2m + l) \end{aligned} \quad (4.35)$$

which proves that $x(k+1)$ is indeed a discrete point, and the same holds trivially true for $v(k+1)$ as well in view of (4.19).

It is now straightforward to apply the discrete SDP algorithm to obtain an optimal closed-loop control law $a(k)^* = R[x(k), k]$, which, for any given vehicle state $x(k)$ and time k , delivers the optimal acceleration $a(k)$. A full vehicle trajectory may also be obtained by starting at an initial state and time and following the optimal encountered acceleration values, until the final state is reached. For illustration, the SDP algorithm may be visualized to run in a 3-D grid, as shown in Fig. 4.2. The grid extends in time over the horizon $[0, k_{\max}]$ and the two-dimensional state space with position x in $[x_{\min}, x_{\max}]$ and speed v in $[v_{\min}, v_{\max}]$. Each node corresponds to a discrete state point of $x(k)$, and each connecting line corresponds to a discrete control

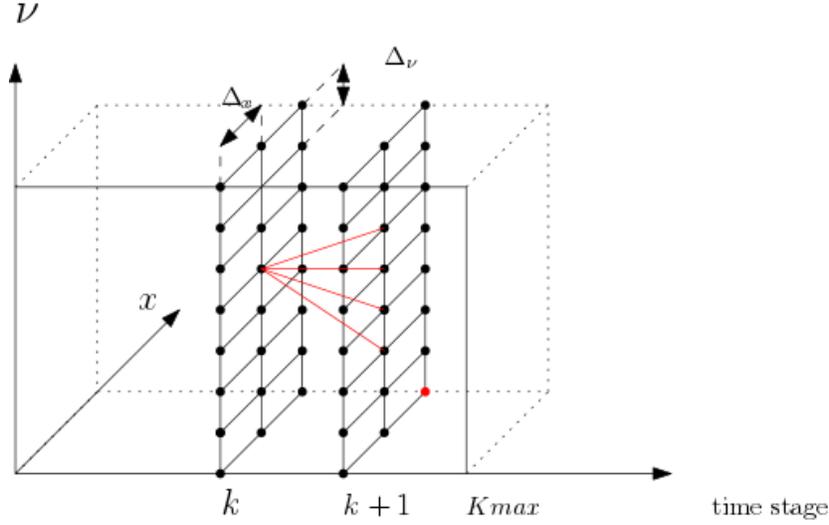


Figure 4.2: Illustration of the 3-D grid of time and state space.

$a(k)$ that leads from $x(k)$ to $x(k+1)$ according to the state equations. The SDP algorithm executes essentially (4.30), and the algorithmic steps are summarized below. Note that the algorithm needs to consider only the case $\tilde{x}(k) = 1$, therefore any arguments pertaining to $\tilde{x}(k)$ are suppressed for convenience.

The SDP algorithm is described as follows:

- 1: $V[\mathbf{x}(k_{\max}), k_{\max}] \leftarrow 0 \quad \forall \mathbf{x}(k_{\max}) \in \mathbf{X}$
- 2: **for** each $k = k_{\max} - 1, \dots, 0$ **do**
- 3: **for** each discrete state $\mathbf{x}(k) \in \mathbf{X}$ **do**
- 4: **for** each discrete control $a(k) \in U$ **do**
- 5: Calculate $x(k+1), v(k+1)$
- 6: **if** $\mathbf{x}(k+1) \notin \mathbf{X}$ **then**
- 7: $V[\mathbf{x}(k), a(k), k] \leftarrow \infty$
- 8: **continue**
- 9: **end if**
- 10: $V[\mathbf{x}(k), a(k), k] \leftarrow \frac{1}{2}a(k)^2 + p(0|k) \cdot J_{DG}^*[\mathbf{x}(k), a(k), k]$
 $\quad \quad \quad + [1 - p(0|k)] \cdot V[\mathbf{x}(k+1), k+1]$
- 11: **end for**
- 12: $V[\mathbf{x}(k), k] \leftarrow \min J[\mathbf{x}(k), a(k), k] \quad \forall a(k) \in U$
- 13: $R[\mathbf{x}(k), k] = a(k)^* \leftarrow \arg \min_{a(k) \in U} \{V[\mathbf{x}(k), a(k), k]\},$

```
14:         with  $a(k)^*$  the optimal control of point  $[\mathbf{x}(k), k]$ 
15:     end for
16: end for
```

4.4.4 Discussion

In the above formulation, there is again (as in the deterministic case) an implicit assumption that a red light is active when the vehicle first appears on the link (at time 0), which is the typical case. The approach, however, can be generalized to the case where the traffic signal is green at the initial time, and its duration is uncertain, but probabilistic information is available regarding the switching time from green to red. The generalized problem (and its solution) may be built in a similar way as in this section, albeit using a longer time horizon, which includes two periods of switching uncertainty: one from red to green (as in the solution above), and an earlier one, reflecting the uncertain switching from green to red. The corresponding details will be published at a future occasion.

As will be reported in the next section, the SDP algorithm may take several minutes to execute in an ordinary PC. This implies that this solution cannot be obtained on the vehicle side in real time. However, the SDP solution is very comprehensive, as it provides optimal acceleration for all feasible vehicle positions and speeds. Therefore, the SDP algorithm may be executed off-line on the infrastructure side, and the signal controller may communicate to approaching vehicles optimal trajectories according to their current position and speed. More specifically, each time a vehicle approaches the traffic signal, several communication messages need to be exchanged between the vehicle and the signal controller, as follows. First, the vehicle sends to the signal controller its position and speed; given this information, the controller sends back the optimal trajectory that the vehicle should follow. This procedure can be applied repeatedly if substantial deviations occur with respect to the vehicle's trajectory, for example, in case of impeding traffic or other disturbances, in such cases, the vehicle may send a new message to the controller with its new position and speed to receive an updated trajectory. At last, when the traffic signal turns green, the vehicle applies the deterministic solution to reach the final state, which can be calculated online in the vehicle. As an alternative approach, the authors are working on alternative solution algorithms, which would just deliver optimal trajectories for each vehicle by solving only a small portion of the above problem (just for the specific vehicle) in much shorter computation time; such algorithms would then be executable in real time, even in an MPC mode, on the vehicle side, similarly to the deterministic GLOSA.

4.5 Results

4.5.1 Investigated Scenarios

In this section, typical application results of the deterministic and the stochastic GLOSA approaches that were discussed in the previous sections are presented. Several scenarios, with different initial and terminal conditions have been tested (Kalogianni, 2018). However, because of limited space, only two scenarios will be presented here, with the respective setups provided below. In both scenarios, the initial and final position and the traffic signal position and switching time are the same, but the initial vehicle speed differs. As a consequence, in Scenario 1, the vehicle's optimal trajectory resulting from UP does not interfere with the red-light phase, while, in Scenario 2, the UP solution violates the green-light constraint, therefore the CP solution delivers the vehicle trajectory, which guides the vehicle to pass the traffic signal location exactly by the time that the traffic light turns green from red.

Scenario 1. $x_0 = 0$ m, $x_1 = 150$ m, $x_e = 220$ m, $v_0 = 5$ m/s, $v_e = 11$ m/s, $t_1 = 18$ s, $w = 0.1$

Scenario 2. $x_0 = 0$ m, $x_1 = 150$ m, $x_e = 220$ m, $v_0 = 11$ m/s, $v_e = 11$ m/s, $t_1 = 18$ s, $w = 0.1$

4.5.2 Deterministic GLOSA

The resulting trajectories of acceleration, speed, and position of the vehicle for both Scenarios 1 and 2, are presented in Fig. 4.3, a and b. In Scenario 1, the initial speed is lower than the targeted final speed, so the vehicle needs to accelerate to fulfill the final condition. In this scenario, the optimal trajectories resulting from UP happen to lead the vehicle to reach the traffic light during the green phase, therefore there is no need to re-adjust the trajectories via solution of CP. On the other hand, in Scenario 2, the initial and final speeds are the same, which means that the only reasons to change the vehicle speed are:

- increase and eventually decrease the speed to reach the fixed final position earlier, that is, decrease the optimal final time; or
- decrease and eventually increase the speed because, with the given initial speed, the vehicle would cross the traffic signal during the red phase.

It may be seen in Fig. 4.3, a and b, that the solution of UP behaves according to the first reason, but fails to satisfy the green-light constraint. In contrast, the CP solution recognizes the second reason and leads the vehicle to decelerate, to pass the traffic light optimally by the time the traffic light turns green, and then accelerate again to fulfill the final conditions. In both scenarios, the target speed v_e and the target position x_e are achieved at the final time t_e as

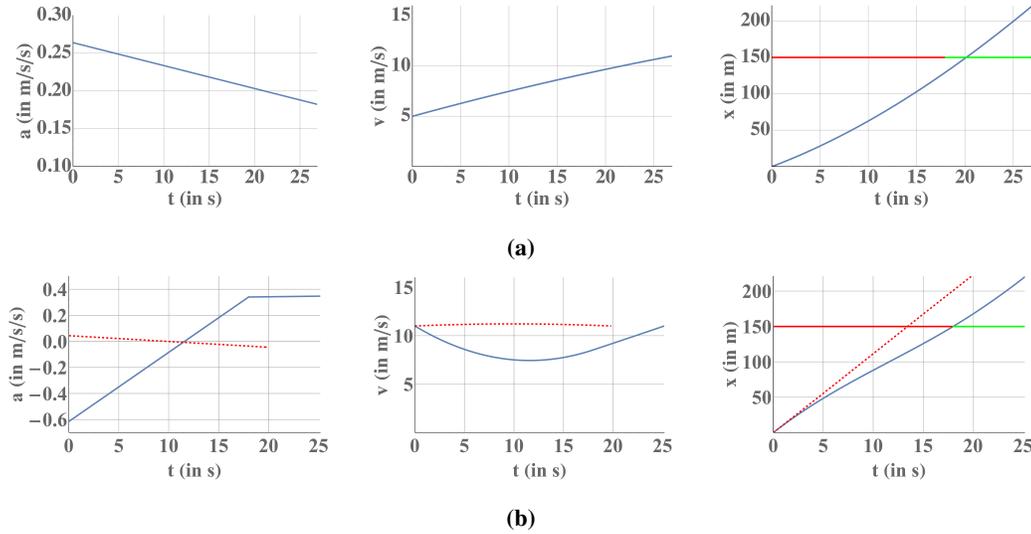


Figure 4.3: Graphical representation of the optimal trajectories for: (a) Scenario 1 and (b) Scenario 2 (blue lines), along with the corresponding trajectories of the UP solution (red dotted lines). Also, the traffic light phases are represented as thick red and green lines for red and green phases, respectively.

requested, and the traffic light constraint is not violated. The resulting optimal final time and cost for the two scenarios are 26.9 s and 2.02 for Scenario 1 and 25.1 s and 2.54 for Scenario 2, respectively. Note also that the trade-off curves displayed in Fig. 4.1 resulted with the set-up of Scenario 1 and may be used to assess the impact of the weight w .

4.5.3 Stochastic GLOSA

The results of the proposed SDP approach are presented in this section. For this case, Scenario 2 is considered, and, in addition, a variation of this scenario (Scenario 2a) is also considered, where the initial position is changed to $x_0 = 50$ m, that is, considering the vehicle closer to the traffic signal. The states and control bounds are set to $[x_{\min}, x_{\max}] = [0, 150]$ m, $[v_{\min}, v_{\max}] = [0, 16]$ m/s, $[a_{\min}, a_{\max}] = [-3, 3]$ m/s², respectively. The time step T is 1 s, which verifies the discretization properties mentioned earlier. The switching time range for the traffic light is $[k_{\min}, k_{\max}] = [10, 30]$ with uniform a-priori probability distribution.

As far as the discretization is concerned, $\Delta = \Delta a = \Delta v = 0.125$ is used, which leads to $\Delta x = 0.0625$ m. The choice of the discretization can be justified by the results of various

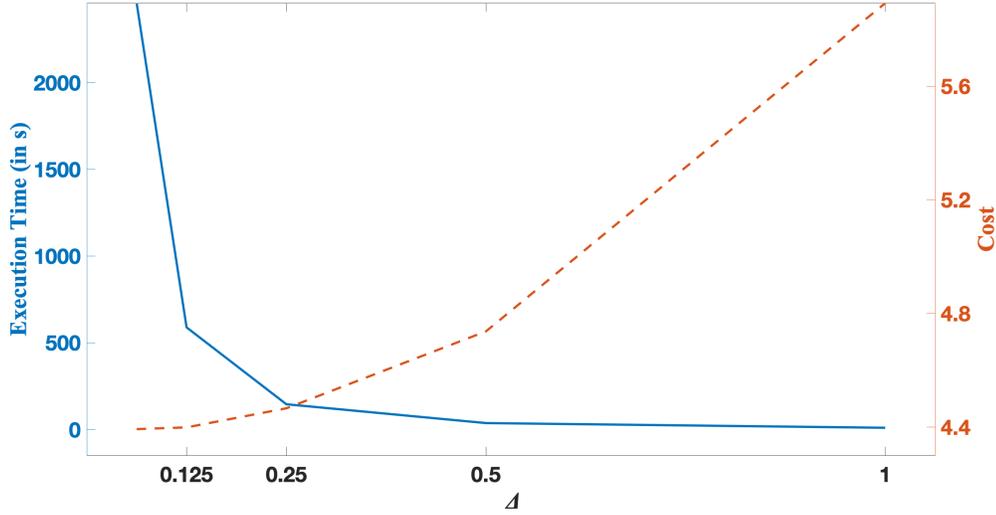


Figure 4.4: Cost (red dotted line) and execution time (blue line) versus discretization interval Δ .

different discretization intervals reported in Fig. 4.4. It can be observed that, as expected, as the discretization interval is reduced, the obtained optimal cost is reducing, but the CPU-time is increasing, since the number of grid points and the number of grid point connections increases accordingly. For the discretization interval $\Delta = 0.125$, the cost has virtually converged to the best achievable outcome, while the CPU-time is not prohibitive (588 s).

For comparison, an alternative (sub-optimal) GLOSA approach for the uncertain switching time case is also investigated, which is as follows:

- Solve the deterministic GLOSA problem under the pessimistic assumption that the traffic light will switch at the latest possible time, that is, at $t_1 = Tk_{\max}$, so as to be on the safe side. (For consistency, the deterministic GLOSA problem was solved here using deterministic dynamic programming, rather than using the analytical solution.)
- Let the vehicle drive accordingly.
- When the switching time t_1 becomes known, switch (on the fly) to the corresponding deterministic GLOSA solution.

Fig. 4.5, present the optimal trajectories of states and control obtained by the SDP approach, compared with the sub-optimal approach described above, for Scenario 2 (see Fig. 4.5a) and

Scenario 2a (see Fig. 4.5b). The trajectories shown are for a single realization of traffic light switching, namely that the traffic light switches from red to green at the latest possible time, which is k_{\max} . Clearly, this is the worst possible case for the SDP approach (which expects in vain that the traffic light might switch earlier), and the best possible case for the sub-optimal approach, which, for this particular realization, turns out to be in fact optimal.

As can be seen in Fig. 4.5, a and b, for both Scenarios 2 and 2a, the SDP vehicle starts moving closer to the traffic signal short before the time k_{\min} . This behavior is rational, as the stochastic approach is aware that there is a probability for the traffic light to switch earlier, therefore the vehicle moves to positions closer to the traffic signal; of course, this behavior entails the risk to have worse transition cost if the traffic light switches late, as in the presented realization. This is even more pronounced for Scenario 2a, in which the SDP vehicle even stops for a short while at the traffic light, which is still red (against the anticipated probabilities).

Despite the outcome of this specific realization, the SDP approach is better informed, as it incorporates all the available knowledge, including the probabilistic distribution of the switching time and its updates over time, as well as the actual traffic light switching being communicated to the vehicle when decided. The SDP approach optimizes the expected performance subject to all this information, but this of course implies that, in specific realizations, its performance cannot be superior to a hypothetical deterministic solution that has exact knowledge of the switching time. The superiority of the SDP approach becomes evident from the bottom-right diagrams in Fig. 4.5, a and b, each of which displays two pairs of curves. The first couple of curves represents the square-of-acceleration cost resulting from each of the two approaches (stochastic-optimal and sub-optimal approaches) versus the actual switching time in the range k_{\min}, k_{\max} . The second couple of curves represents the corresponding fuel consumption for the two approaches, calculated with the well-known Australian Road Research Board (ARRB) fuel consumption model (Akçelik, Biggs, 1987). Specifically, the displayed acceleration cost and fuel consumption values comprise the acceleration cost and fuel consumed from the initial time up to the actual switching time, plus the escape cost and the corresponding fuel consumption thereafter—in dependence of the actual switching time. It may be seen that the SDP approach achieves the best possible outcome and the biggest improvement over the sub-optimal approach for actual switching at k_{\min} . This improvement gradually diminishes as the actual switching time increases, and there is a cross-point toward bigger switching times, beyond which the sub-optimal approach becomes better, as the switching time approaches the assumption of that approach. The biggest difference in favor of the sub-optimal approach occurs, as expected, at k_{\max} , where, as mentioned, the sub-optimal approach is deterministically optimal. Overall, given the uniform probabilistic distribution, it is evident by inspection of both couples of curves, that the SDP approach is superior (in fact, optimal on average) as the inter-curve surface before the cross-point of both curves is clearly bigger than after, and this applies to both curve couples.

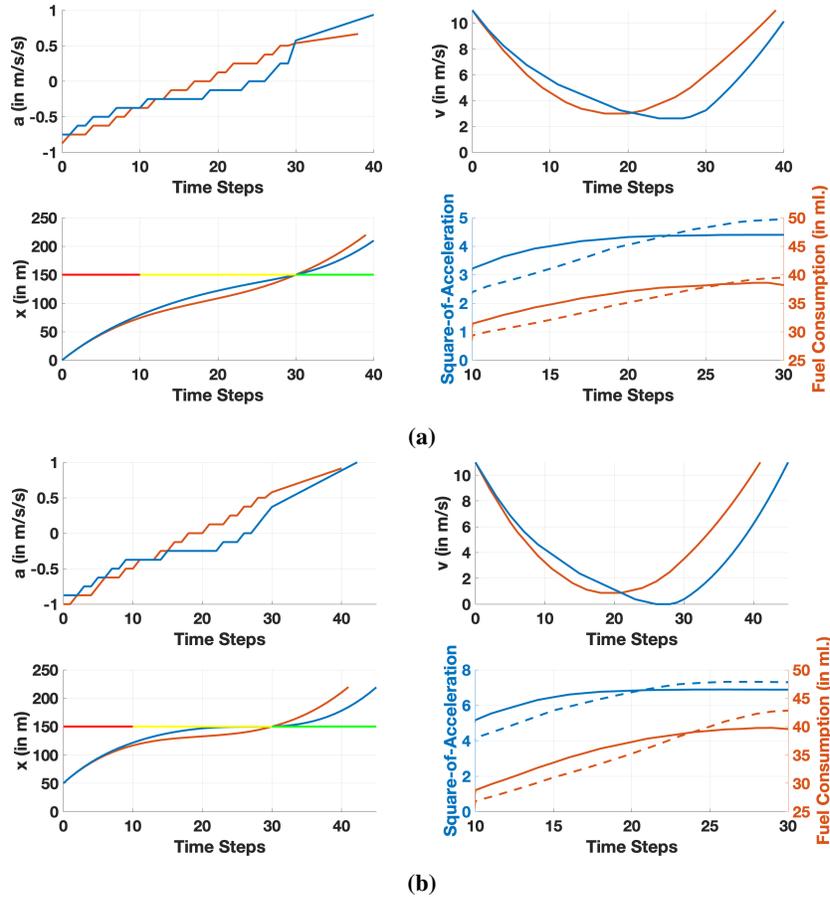


Figure 4.5: Optimal states and control trajectories of SDP (blue lines) compared with those of the sub-optimal approach (red lines) for: (a) Scenario 2 and (b) Scenario 2a. The traffic light phases are represented as a straight line which is red for the red phase, yellow for the stochastic switching period, and green for the rest. The bottom-right diagram in each scenario displays the resulting square-of-acceleration cost (blue lines) and the fuel consumption (red lines), the latter based on the Australian Road Research Board (ARRB) model, for each approach (continuous lines for the SDP and dashed lines for the sub-optimal approach) versus the actual switching time in the range $[k_{\min}, k_{\max}]$.

The general superiority of the SDP approach is verified in Fig. 4.6a, which displays the average percentage difference of fuel consumption (calculated with the ARRB model) of SDP

versus the sub-optimal approach for different initial state values x_0, v_0 . It can be seen that SPD achieves always a (smaller or bigger) fuel reduction, compared with the sub-optimal approach. Specifically, for the setup of Scenario 2 (Fig. 4.6a), the average percentage difference is quite noticeable either when the vehicle is far from the traffic signal with high initial speed; or close to the traffic light with low speed; while it is insignificant when the vehicle starts closer to the traffic light with high speed values. This observation is rational as, in the first case, the vehicle has the space and time to move closer to traffic signal as advised; while in the second case, where the space is limited, the trajectories of the two approaches are quite similar. Fig. 4.6b displays the results of another problem setting, which differs from the previous one in that the time-window of possible switching times is enlarged to $[40, 60]$ s; in addition, a bigger time horizon and larger distance from the traffic light are considered in the figure. It may be seen that even higher percentage differences are achieved in this case for some initial states.

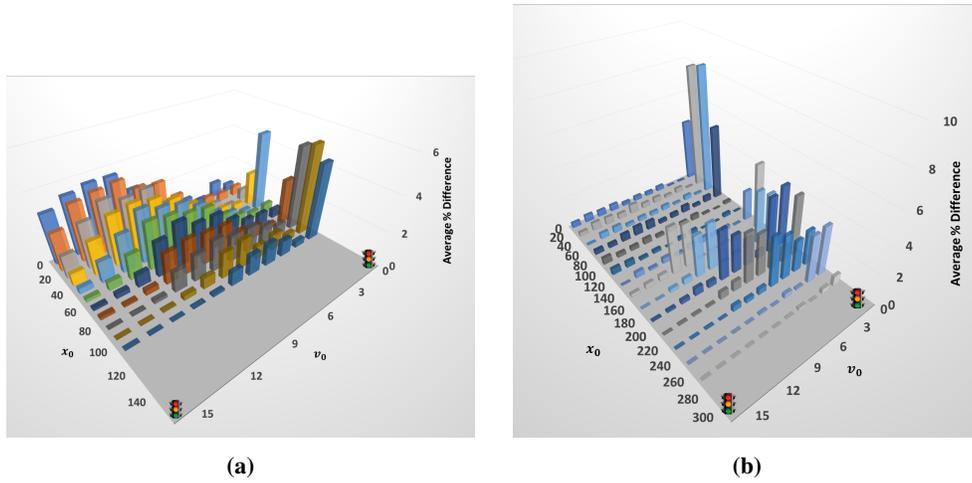


Figure 4.6: Average percentage difference of fuel consumption of SDP versus the sub-optimal approach, calculated with the ARRB fuel consumption model, for different initial state values x_0 (in m), v_0 (in m/s) for: (a) the setup of Scenario 2 and (b) same setup but different time-window of switching times ($k_{\min}, k_{\max} = [40, 60]$) with extended time horizon and distance from the traffic signal ($x_{\max} = 300$).

4.6 Conclusion

A deterministic and a stochastic GLOSA methodology were developed, as well as a sub-optimal base-case approach for the stochastic case, by optimizing vehicles' kinematic trajectories subject to the intermediate traffic signal constraints and with a fixed final state that allows for commensurable consideration of all possible trajectories before picking the optimal one. The GLOSA problem was formulated as an optimal control problem and solved, firstly, analytically, for the case of known switching time of the traffic light, and then numerically through SDP techniques, for the case of uncertain switching time. In the latter case, the traffic light's switching time is assumed to be stochastic, with a given a-priori probability distribution within a known time range $[k_{\min}, k_{\max}]$, which is appropriately updated as time progresses. The SDP approach takes advantage of the analytical solution, as it solves the problem up to the time that the traffic light switches to green, and then uses the analytical solution to compute the rest of the trajectory up to the vehicle's final state downstream the traffic light. By this, the advantages of the SDP and the analytical solution are combined to reduce the complexity and the computational time.

Demonstration results confirm that vehicles guided by the proposed SDP approach achieve better average performance compared with the vehicles guided by the sub-optimal approach, which assumes first that the traffic light switching time is the maximum one and switches eventually to an escape path, when the traffic light switches to green. The superiority of the SDP approach is because it is better informed, incorporating all the available knowledge, including the probabilistic distribution and its update for the traffic light switching time, as well as knowledge that the traffic light switching will be communicated to the system in real time. It should be noted that the average superiority of the SDP approach does not exclude specific instances for which the sub-optimal approach performs better.

The computation of the SDP solution for the whole state-time area of a junction takes some minutes, but may take place off-line, as it produces a complete closed-loop solution that may be used to determine any optimal vehicle trajectories in that area. Ongoing work addresses the solution of parts of the SDP problem, which may reduce the related computation time sufficiently to enable its execution within the vehicle in real time.

Chapter 5

Modified Dynamic Programming Algorithms for GLOSA Systems with Stochastic Signal Switching Times

5.1 Abstract

A discrete-time stochastic optimal control problem was recently proposed to address the GLOSA (Green Light Optimal Speed Advisory) problem in cases where the next signal switching time is decided in real time and is therefore uncertain in advance. The corresponding numerical solution via SDP (Stochastic Dynamic Programming) calls for substantial computation time, which excludes problem solution in the vehicle's on-board computer in real time. To overcome the computation time bottleneck, as a first attempt, a modified version of Dynamic Programming, known as Discrete Differential Dynamic Programming (DDDP) was recently employed for the numerical solution of the stochastic optimal control problem. The DDDP algorithm was demonstrated to achieve results equivalent to those obtained with the ordinary SDP algorithm, albeit with significantly reduced computation times. The present work considers a different modified version of Dynamic Programming, known as Differential Dynamic Programming (DDP). For the stochastic GLOSA problem, it is demonstrated that DDP achieves quasi-instantaneous (extremely fast) solutions in terms of CPU times, which allows for the proposed approach to be readily executable online, in an MPC (Model Predictive Control) framework, in the vehicle's on-board computer. The approach is demonstrated by use of realistic examples. It should be noted that DDP does not require discretization of variables, hence the obtained solutions may be slightly superior to the standard SDP solutions.

5.2 Introduction

In an era of climate crisis, actions toward environment protection are due in Transportation systems, which have a significant impact on environmental pollution and energy resources shortage. Improved fuel efficiency in road vehicles implies reduced fuel consumption, emissions and expenses for the driver. Efficient intelligent transportation systems, including real-time traffic signals, contribute to this effort. Traffic lights were introduced, in the first place, as a measure to ensure safe crossing of antagonistic streams of vehicles, cyclists and pedestrians at urban junctions. However, enforcing safety via traffic lights affects the fuel consumption, as some vehicles are obliged to fully stop in front of a red light; and then accelerate, after the traffic light switching to green. To reduce the resulting vehicle delays and number of stops, several algorithms have been proposed and deployed over the past decades, aiming at optimizing the traffic signals operation (Hounsell, McDonald, 2001; Kosmatopoulos et al., 2006; Papageorgiou et al., 2003). In this context, fuel consumption is increasingly considered as an optimization or evaluation criterion, while developing and deploying signal control systems (Jamshidnejad et al., 2017).

Fixed-time signal plans are derived offline based on historical constant demands; and are applied without deviations. This implies that switching times of the traffic lights are known in advance. In contrast, real-time signal control strategies make use of real-time measurements to compute in real time suitable signal settings. Depending on the employed signal control strategy, the control update period may range from one second to one signal cycle. Clearly, for real-time signals, the next switching time is not known before the switching decision has been actually made.

A common dilemma, for a vehicle approaching a red traffic light, is whether it should slow down or maintain its speed at risk to stop if the traffic light has not switched to green at arrival. This dilemma may be addressed by appropriately designed on-board systems, which receive the current state and timing of the traffic signal and guide the driver (or an automated vehicle), while approaching the traffic light, by giving speed advice that ensures that the vehicle will cross the traffic signal at green and with minimum fuel consumption and emissions. Systems (or apps) optimizing the vehicle approach to traffic lights are often referred to as Green Light Optimal Speed Advisory (GLOSA) systems (Stahlmann et al., 2016).

In the case of fixed signals, and hence prior knowledge of the next switching time, a corresponding message is broadcasted by the signal controller well before switching. Under these conditions, the problem of how to optimize the approach to traffic signals has been addressed in different ways, including rule-based algorithms (Katsaros et al., 2011; Ma et al., 2018; Sanchez et al., 2006) and optimal control approaches, which consider explicitly the vehicle kinematics and are, by their nature, more efficient in producing fuel-optimal speed profiles (Lawitzky et al., 2013; Typaldos et al., 2020a). Some of these works take into

consideration additional factors, such as the car waiting queue. In (Yang et al., 2016), an eco-CACC (Cooperative Adaptive Cruise Control) scheme was developed, where optimal speed trajectories were calculated by minimizing fuel-consumption and ensuring that the vehicles arrive at the intersection as soon as the last queued vehicle is discharged. An extension of this work for the case of multiple signalized intersections is presented in (Yang et al., 2020). In the literature, most existing GLOSA problems are focusing on developing strategies for a single vehicle, however, there are few works that approach the problem in case of platoons of vehicles in mixed traffic environment. In (Zhao et al., 2018), a model predictive control (MPC) approach was proposed that minimizes the fuel-consumption of a platoon of mixed vehicles (automated and human-driven vehicles) while crossing an intersection during the green phase.

On the other hand, the case of real-time signals with very short (e.g., second-by-second) control update periods is more complicated, as there is no exact prior knowledge of the next switching time. In this case, the best available knowledge can be presented as an estimate or as a probabilistic distribution for the next switching time within a short-term future time-window. A comparison of several signal control methods with respect to traffic efficiency, plan stability and the resulting GLOSA performance is presented in (Blokpoel, Lu, 2017). In (Koukoumidis et al., 2011), a system called SignalGuru was developed, which detects and predicts the traffic signal timing using a collection of mobile phone data. Instead of a single estimate, Mahler, Vahidi (2012) produced a probability distribution for the next switching time based on past signal statistics. However, the available probability distribution is used heuristically (rather than optimally) to accordingly time-weight the objective function within a deterministic optimal control problem that is solved via a dynamic programming (DP) algorithm. In (Sun et al., 2020), the uncertainty of traffic signal timing is considered as a data-driven chance-constrained robust optimization problem, where the red-light duration is defined as a random variable. A data-driven approach is adopted to formulate chance constraints, based on empirical sample data, and Dynamic Programming is employed to solve the optimization problem. In (Lawitzky et al., 2013), a stochastic optimal control problem was proposed, which uses a probabilistic distribution of the next signal switching time, and the problem is solved using a discrete stochastic dynamic programming (SDP) algorithm.

In (Typaldos et al., 2020a), the problem of producing fuel-optimal vehicle trajectories for a vehicle approaching a traffic signal was considered for both cases of known and stochastic switching times. For the first case, an optimal control problem was formulated and solved analytically via PMP (Pontryagin's Maximum Principle). Subsequently, the case of stochastic switching time with known probability distribution was also addressed in the format of a stochastic optimal control problem, which was solved numerically using SDP. The proposed SDP algorithm may take several minutes to execute, which implies that the solution is not real-time feasible and can therefore not be obtained on-board the vehicle, but must be

executed offline, on the infrastructure side, and be communicated to approaching vehicles according to their current states. To substantially reduce the computation time and memory requirements for the solution of the stochastic GLOSA problem and enable its solution on-board the vehicle, Typaldos et al. (2021) employed a Discrete Differential Dynamic Programming (DDDP) algorithm (Heidari et al., 1971). DDDP is an iterative algorithm, each iteration receiving a feasible (but non-optimal) trajectory and transforming it to an enhanced one, to be used in the next iteration, and so forth, until convergence to the overall solution is achieved. Each iteration uses the conventional DP algorithm to solve the problem within a strongly reduced state domain around the received state trajectory of the previous iteration. The computational time required for each iteration is far lower compared to that of the one-shot full problem solution, due to the strongly reduced space domain considered. Thus, the problem may be solved by use of DDDP very fast (around 1 s CPU time or less) (Typaldos et al., 2021). It should be noted that, in contrast to deterministic optimal control problems, general stochastic optimal control problems do not feature a solution trajectory, even for given initial states, due to the uncertainty of system evolution created by the included stochastic variables. However, in the specific stochastic GLOSA problem, such a trajectory is indeed present in the problem solution, and this allows for application of the DDDP algorithm.

As an attempt to further reduce the computation time for the solution of the stochastic GLOSA problem and enable its solution on-board the vehicle, a Differential Dynamic Programming (DDP) algorithm is employed in the current work. DDP was first introduced in (Jacobson, Mayne, 1970), while several extensions and applications were eventually conducted by Murray, Yakowitz (1979, 1984). DDP is also used in robotics (Budhiraja et al., 2018; Tassa et al., 2014) and unmanned aerial vehicles (UAV) (Xie et al., 2017). DDP is also an iterative algorithm, which starts with a user-defined feasible trajectory to be used in the first iteration. Each iteration produces an improved trajectory, to be used in the subsequent iteration; until convergence to the solution of the original optimal control problem is achieved. Each iteration solves, for each time-step separately, a quadratic-linear approximation of the stochastic recursive Bellman equation; the approximation being computed around the initial trajectory of the iteration. It should be noted that, in contrast to the standard DP and DDDP algorithms, DDP requires no discretization of the state and control spaces, hence it may result in better solutions compared with discretised-value formulations. Moreover, DDP is found to deliver solutions quasi-instantaneously, which enables the approach to be readily implementable in an MPC (model predictive control) framework online, on the vehicle's computer. Note that communication between vehicles and traffic signals (V2I) is crucial in GLOSA problems (Lu et al., 2020), however in this work we focus on the methodological part of such systems.

The remainder of the paper is organized as follows: in Section 5.3, the optimal control problems with known signal switching time and uncertain signal switching time, as proposed

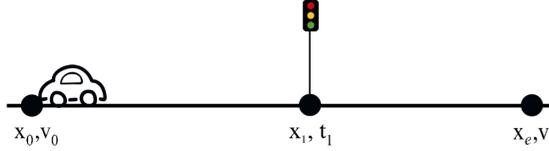


Figure 5.1: A vehicle starting from a given initial state (x_0, v_0) aims to reach a fixed final state (x_e, v_e) in presence of a traffic signal at a given position x_1 that switches to green at time t_1 .

in (Typaldos et al., 2020a) are briefly presented for completeness. Section 5.4 presents the DDDP algorithm, along the lines of (Typaldos et al., 2021). Section 5.5 presents the DDP-based GLOSA problem algorithm, while demonstration results for the DDP algorithm’s performance and comparison with the one-shot stochastic GLOSA and the DDDP results are presented in Section 5.6. Finally, Section 5.7 concludes this work, summarizing its contributions.

5.3 Optimal Control-Based GLOSA with Known or Unknown Signal Switching Time

This section outlines, for completeness, the deterministic and stochastic GLOSA optimal control problems, for the cases of known and unknown signal switching time, respectively, as well as the SDP solution algorithm for the latter, following (Typaldos et al., 2020a).

5.3.1 Known Signal Switching Time - Deterministic GLOSA

Consider a vehicle traveling from an initial state $\mathbf{x}_0 = [x_0, v_0]^T$, x_0 being a given initial position and v_0 a given initial speed, with the purpose to reach a fixed final state $\mathbf{x}_e = [x_e, v_e]^T$ within a free (but weighted) time horizon t_e . Between the initial and final positions, there is a traffic signal, and hence the additional restriction that the vehicle must not pass the signal position x_1 before the known time t_1 , when the traffic light turns green from red (see Fig. 5.1). The implicit assumption here is that a red light is active when the vehicle appears on the link (at time 0), but a generalisation, which includes the case where the vehicle appears when the traffic light is green, is given in (Typaldos et al., 2020a). The vehicle task is to appropriately adjust its acceleration (control variable), so as to minimize fuel consumption, while satisfying the initial and final conditions \mathbf{x}_0 and \mathbf{x}_e , as well as the intermediate (traffic signal) constraint.

The minimization problem outlined above is formulated as an optimal control problem, which accounts for the vehicle kinematics via the following state equations:

$$\dot{x} = v \quad (5.1)$$

$$\dot{v} = a \quad (5.2)$$

where a is the vehicle acceleration, which assumes the role of the control variable. The objective criterion to be minimized reads

$$J = w \cdot t_e + \frac{1}{2} \int_0^{t_e} a^2 dt \quad (5.3)$$

The utilized acceleration cost term a^2 in the criterion was demonstrated to be an excellent proxy for deriving fuel-minimizing vehicle trajectories (Typaldos et al., 2020b). The final time t_e is free but penalized with the weight w . For higher values of w , the resulting t_e will be lower and vice versa. This, consequently, affects the acceleration cost, which, depending on higher or lower w value, will also have increased or decreased values (for more details, see (Typaldos et al., 2020a)). In addition, the green-light constraint, $t_S \geq t_1$, must be fulfilled, where t_S is the time at which the vehicle crosses the signal position x_1 , that is, $x(t_S) = x_1$. If necessary, upper and lower bounds may be applied to speed v and acceleration a .

The solution of this problem was addressed in (Typaldos et al., 2020a) and can be obtained analytically using symbolic differentiation tools. Thus, the numerical solution of the deterministic GLOSA problem, for a specific problem instance, takes only fractions of a second of computation time and can be executed in real time on-board for each approaching vehicle. Note that it may be reasonable to continuously update the vehicle trajectory (in a model predictive control (MPC) loop) to account for possible deviations from the first calculated vehicle trajectory, which may occur because of a variety of disturbances, including a slower vehicle ahead.

For a given junction, the final state is the same for any initial vehicle state \mathbf{x}_0 and any switching time t_1 . Therefore, the optimal value of criterion (5.3) of the deterministic GLOSA solution depends only on the initial state and the switching time and is denoted $J_{DG}^*(\mathbf{x}_0, t_1)$ for later use.

5.3.2 Uncertain Signal Switching Time Problem – Stochastic GLOSA

The traffic light switching time may be subject to short-term decisions in dependence of the prevailing traffic conditions in cases of real-time signals. In such cases, we typically have minimum and maximum admissible switching times; hence, based on statistics from past signal switching activity, we may derive a probability distribution of switching times within

the admissible time-window of possible signal switching times. Thus, the problem can be cast in the format of a stochastic optimal control problem, which may be solved numerically using SDP techniques. To this end, the analytical solution of the deterministic GLOSA optimal control problem is used within the stochastic approach, as will be explained in this section (for more details, see (Typaldos et al., 2020a)).

For the stochastic optimal control problem and the SDP algorithm (Bertsekas, 2012), the discrete-time version of the vehicle kinematics, with time step T , is considered, as follows:

$$x(k+1) = x(k) + v(k)T + \frac{1}{2}a(k)T^2 \quad (5.4)$$

$$v(k+1) = v(k) + a(k)T \quad (5.5)$$

where $x(k)$ and $v(k)$ correspond to the vehicle position and speed, respectively, at discrete times $k = 0, 1, \dots$ (where $kT = t$), while the control variable $a(k)$ reflects the acceleration that remains constant over each time period k . The state and control variables are bounded within the following feasible regions

$$\mathbf{x}(k) \in \mathbf{X} = [\mathbf{x}_{min}, \mathbf{x}_{max}] \quad (5.6)$$

$$a(k) \in U = [a_{min}, a_{max}] \quad (5.7)$$

with $\mathbf{x}_{min}, \mathbf{x}_{max}$ and a_{min}, a_{max} being the lower and upper bounds of the states and acceleration, respectively. The traffic light's discrete switching time k_1 is not known, but it is assumed that a known range $k_{min} \leq k_1 \leq k_{max}$ of possible switching times exists, k_{min} and k_{max} being the minimum and maximum possible switching times, respectively.

For proper problem formulation, a virtual state $\tilde{x}(k)$ is introduced, that reflects formally the stochasticity of traffic light switching

$$\begin{aligned} \tilde{x}(k+1) &= \tilde{x}(k) \cdot z(k) \\ \tilde{x}(0) &= 1 \end{aligned} \quad (5.8)$$

where $z(k)$ is a binary stochastic variable defined as

$$z(k) = \begin{cases} 0 & \text{if traffic light switches at time } k+1 \\ 1 & \text{else} \end{cases} \quad (5.9)$$

with (5.8) and (5.9), the virtual state $\tilde{x}(k)$ is either equal to 1 if the traffic light has not yet switched until time $k-1$; or equal to 0 if switching occurred at time k or earlier. The virtual state \tilde{x} is assumed measurable, which means that the system knows at each time kT if switching

has taken place or not within the last time period $[(k-1)T, kT]$.

The stochastic variable $z(k)$ is independent of its previous values and takes values according to a time-dependent probability distribution $p(z|k)$. Based on the statistics of previous signal switching activity, availability of an a-priori discrete probability distribution $P(k), k_{min} \leq k \leq k_{max}$, is assumed, for signal switching within the time-window, where $\sum_{k=k_{min}}^{k_{max}} P(k) = 1$. Since no switching takes place for $k \leq k_{min} - 1$, we have

$$p(0|k) = 0 \quad \text{for } k < k_{min} - 1 \quad (5.10)$$

For $k \geq k_{min}$, the probability distribution $p(z|k)$, is obtained by use of "crop-and-scale", meaning that the a-priori probabilities of previous time steps, where switching did not take place, are distributed analogously to increase the probabilities of the remaining discrete times within the time-window (Lawitzky et al., 2013). As shown in (Typaldos et al., 2020a), this update may be done by use of the following crop-and-scale formula that applies for $k_{min} \leq k \leq k_{max} - 1$ and for any a-priori distribution $P(k)$

$$p(0|k) = \frac{P(k+1)}{\sum_{\kappa=k+1}^{k_{max}} P(\kappa)} \quad (5.11)$$

The cost criterion of the stochastic problem is essentially the same as (5.3) in the deterministic case. However, in the stochastic case, the exact value of the criterion depends on the stochastic variable's realization, and therefore we consider minimisation of the expected value. As shown in (Typaldos et al., 2020a), the cost criterion in discrete time becomes

$$J = E \left\{ \frac{1}{2} \sum_{k=0}^{k_1-1} a(k)^2 + J_{DG}^*[\mathbf{x}(k_1), k_1] \right\} \quad (5.12)$$

where the expectation refers to the stochastic variable $z(k), k = 0, \dots, k_{max} - 1$. This criterion captures the quadratic acceleration cost up until the stochastic switching time k_1 , when the vehicle is at state $\mathbf{x}(k_1)$. After the signal switching, the problem instantly becomes a deterministic GLOSA problem (Section 5.3.1), and the corresponding optimal cost-to-go is $J_{DG}^*[\mathbf{x}(k_1), k_1]$, which will be denoted as the "escape cost".

To obtain a formally proper cost criterion, the stochastic variable $z(k)$ and the virtual variable $\tilde{x}(k)$, introduced earlier, are used, and, as shown in (Typaldos et al., 2020a), this yields the objective function in the required form, as follows

$$J = E \left\{ \tilde{x}(k) \sum_{k=0}^{k_{max}-1} \left[\frac{1}{2} a(k)^2 + [1 - z(k)] J_{DG}^*[\mathbf{x}(k), a(k), k+1] \right] \right\} \quad (5.13)$$

where, for notational simplicity, $J_{DG}^*[\mathbf{x}(k), a(k), k+1] = J_{DG}^*[\mathbf{x}(k+1), k+1]$ when $\mathbf{x}(k+1)$ is replaced via the state equations (5.4), (5.5) as a function of $\mathbf{x}(k), a(k)$.

Equations (5.4)-(5.13) constitute an ordinary stochastic optimal control problem (Bertsekas, 2012). Denoting the corresponding optimal cost-to-go function by $V[\mathbf{x}(k), \tilde{x}(k), k]$, the recursive Bellman equation for $0 \leq k \leq k_{max} - 1$ reads

$$V[\mathbf{x}(k), \tilde{x}(k), k] = \min_{a(k) \in U} \left\{ E \left\{ \frac{1}{2} a(k)^2 + [1 - z(k)] J_{DG}^*[\mathbf{x}(k), a(k), k+1] + V[\mathbf{x}(k+1), \tilde{x}(k)z(k), k+1] \right\} \right\} = \min_{a(k) \in U} \left\{ \frac{1}{2} a(k)^2 + p(0|k) \cdot J_{DG}^*[\mathbf{x}(k), a(k), k+1] + [1 - p(0|k)] \cdot V[\mathbf{x}(k+1), 1, k+1] \right\} \quad (5.14)$$

with boundary condition $V[\mathbf{x}(k_{max}), 1, k_{max}] = 0$. Note that, the minimum is required with respect to the control of time k only, i.e., $a(k) \in U$, as typical in dynamic programming, which facilitates the numerical solution.

In this formulation, it has been assumed that the decision on the traffic light switching is taken within the last time period before the actual switching. The generalization to the case of taking a switching decision κ time periods in advance of the actual switching is easy and is mentioned in (Typaldos et al., 2020a).

5.3.3 Discrete SDP Numerical Solution Algorithm

For the numerical solution of the stochastic problem, using the SDP algorithm, the state and control variables must be discretised. As the discretization level has a significant impact on computational time and memory requirements, but also on the accuracy of the computed solution, an appropriate trade-off must be specified between reasonable computation requirements versus achievable solution quality. For the discretization, the discrete time interval T is set equal to 1 s, which is a reasonable choice for the problem at hand. Then, a general discretization interval Δ for the problem variables is assumed, and the discretization interval of acceleration is set $\Delta a = \Delta$. From (5.5), the discretization interval of speed assumes the same value ($\Delta v = \Delta a = \Delta$). Likewise, in view of (5.4), the discretization interval for the position is

$$\Delta x = \frac{1}{2} \Delta \cdot T^2 \quad (5.15)$$

Based on these settings, it was shown in (Typaldos et al., 2020a) that, if $x(k), v(k), a(k)$ are discrete points, then $x(k+1)$ and $v(k+1)$ resulting from (5.4) and (5.5) are also discrete points.

It is now straightforward to apply the discrete SDP algorithm to obtain a globally optimal closed-loop control law $a^*(k) = R[\mathbf{x}(k), k]$, which, for any given vehicle state $\mathbf{x}(k) \in \mathbf{X}$ and time k , delivers the corresponding optimal acceleration $a^*(k)$. The SDP algorithmic steps are summarized below. Note that the algorithm needs to consider only the case $\tilde{x}(k) = 1$, therefore any arguments pertaining to $\tilde{x}(k)$ are suppressed for convenience.

The SDP algorithm is as follows:

- 1: $V[\mathbf{x}(k_{\max}), k_{\max}] \leftarrow 0 \quad \forall \mathbf{x}(k_{\max}) \in \mathbf{X}$
- 2: **for** each $k = k_{\max} - 1, \dots, 0$ **do**
- 3: **for** each discrete state $\mathbf{x}(k) \in \mathbf{X}$ **do**
- 4: **for** each discrete control $a(k) \in U$ **do**
- 5: Calculate $x(k+1), v(k+1)$
- 6: **if** $\mathbf{x}(k+1) \notin \mathbf{X}$ **then**
- 7: $\Phi[\mathbf{x}(k), a(k), k+1] \leftarrow \infty$
- 8: **continue**
- 9: **end if**
- 10: $\Phi[\mathbf{x}(k), a(k), k+1] \leftarrow \frac{1}{2}a(k)^2 + p(0|k) \cdot J_{DG}^*[\mathbf{x}(k), a(k), k+1]$
 $\quad + [1 - p(0|k)] \cdot V[\mathbf{x}(k+1), k+1]$
- 11: **end for**
- 12: $V[\mathbf{x}(k), k] \leftarrow \min \Phi[\mathbf{x}(k), a(k), k+1] \quad \forall a(k) \in U$
- 13: $R[\mathbf{x}(k), k] = a^*(k) \leftarrow \arg \min_{a(k) \in U} \{\Phi[\mathbf{x}(k), a(k), k+1]\}$,
- 14: with $a^*(k)$ the optimal control of point $[\mathbf{x}(k), k]$
- 15: **end for**
- 16: **end for**

As mentioned, this algorithm delivers a globally optimal control law $a^*(k) = R[\mathbf{x}(k), k]$ for the full state domain \mathbf{X} . Note that general stochastic optimal control problems do not possess a solution trajectory for specific initial states \mathbf{x}_0 , because the state evolution is uncertain in presence of the stochastic variables. However, in the specific stochastic GLOSA problem addressed here, the evolution of the vehicle states (5.4) and (5.5) is not affected by the stochastic variable $z(k)$, which concerns only the signal switching time. Thus, for a given initial state, i.e., vehicle position and speed at time 0, the optimal control law may be used to produce an optimal vehicle trajectory that the vehicle should pursue; until the signal switching actually occurs, in which case the vehicle must instantly behave according to the deterministic GLOSA solution.

Note that the SDP algorithm may take several minutes to execute in an ordinary PC, as will be reported in Section 5.6. This implies that the solution cannot be obtained on the vehicle side in real time.

5.4 Discrete Differential Dynamic Programming (DDDP)

In (Typaldos et al., 2021), a modified DP algorithm called DDDP was employed for the stochastic GLOSA problem in order to address the major disadvantage of the discrete (S)DP algorithm, which is the high computation time required for the numerical solution of the optimal control problem, which increases exponentially with the number of state and control variables (Papageorgiou et al., 2015). The DDDP algorithm was proposed in (Heidari et al., 1971) for deterministic optimal control problems. The method can nevertheless be applied here, because an optimal vehicle trajectory may be derived for the stochastic GLOSA problem, despite its stochastic character, as mentioned in the previous section. The DDDP approach to the stochastic GLOSA problem is presented in this paper in more detail and with more comprehensive testing and comparison results, compared to the conference article (Typaldos et al., 2021).

DDDP is an iterative algorithm, calling for a feasible starting state trajectory to be specified externally. Each iteration l receives a feasible (but non-optimal) state trajectory $\mathbf{x}^{(l-1)}(k)$, and transforms it to an enhanced one $\mathbf{x}^{(l)}(k)$, to be used in the next iteration. To this end, each iteration solves a discrete SDP problem by use of the standard SDP algorithm presented above. What changes at each iteration l is the considered state domain

$$\mathbf{X}_C^{(l)} = \left\{ \mathbf{x}(k) \mid |\mathbf{x}(k) - \mathbf{x}^{(l-1)}(k)| \leq \Delta_C^{(l)} \wedge \mathbf{x}(k) \in \mathbf{X} \right\} \quad (5.16)$$

which is a strongly reduced subdomain of the original state domain \mathbf{X} in (5.6). In other words, the discretized SDP problem is solved in each iteration l within a corridor with width $\Delta_C^{(l)}$ around the received state trajectory $\mathbf{x}^{(l-1)}(k)$ of the previous iteration, to produce a solution trajectory $\mathbf{x}^{(l)}(k)$ for use in the next iteration. The corridor width $\Delta_C^{(l)}$, as well as the discretisation intervals $\Delta a^{(l)}, \Delta \mathbf{x}^{(l)}$, can vary in each iteration, typically at a decreasing rate. The iterations stops when a termination criterion is satisfied.

In the proposed GLOSA application (Typaldos et al., 2021), the starting trajectory for the first iteration of DDDP is the optimal solution of the deterministic GLOSA problem, assuming the "pessimistic" case where the traffic light will switch from red to green at the latest possible time, that is, at $k_1 = k_{max}$, so as to cover the whole time range and be not too far from the stochastic optimal trajectory; see (Typaldos et al., 2020a). The discretisation intervals $\Delta a^{(l)}, \Delta \mathbf{x}^{(l)}$ for the first iteration are also given. The intervals are reduced by half, each time there is no solution improvement at two subsequent iterations. The corridor width is taken as

$$\Delta_C^{(l)} = \mathbf{C} \cdot \Delta a^{(l)} \quad (5.17)$$

where $\mathbf{C} = [C_x, C_v]$ are constant given values specifying the state corridor width. Thus, the

corridor's initial width is defined through the chosen values for \mathbf{C} and $\Delta a^{(l)}$. In the following iterations, whenever the discrete interval is reduced, there is an analogous reduction of the corridor width. Consequently, we have a constant number of feasible discrete points in all iterations, which facilitates the algorithm's fine-tuning, so as to improve its computational efficiency. Note that, C_x and C_v values may differ, as the magnitude of the two respective state variables differs.

The admissible control region U (see (5.7)) is kept the same at each iteration, although many related state transitions cannot be considered in view of the reduced state variables domain. The algorithm terminates whenever there is no improvement of the produced solution trajectory even after a reduction of the discretisation interval; or when the $\Delta a^{(l)}$ value becomes less than 0.125, which was found in (Typaldos et al., 2020a) to lead to sufficiently accurate results.

It should be noted that there is no general guarantee that the DDDP iterations will actually converge to the full-domain SDP solution. In particular, if the state sub-domains $\mathbf{X}_C^{(l)}$ considered in the iterations are too small, the obtained DDDP solution may actually differ from the overall SDP solution. On the other hand, if the state sub-domains are selected large, the required number of iterations may decrease, but the computation time required to find the solution at each iteration increases accordingly. In conclusion, some fine-tuning regarding the size of sub-domains is necessary to ensure convergence to the overall SDP solution with minimum overall (all iterations) computation time.

5.5 Differential Dynamic Programming (DDP)

DDP is an iterative algorithm, which requires a given initial feasible trajectory (of states and controls) to be used in the first iteration. At each iteration and time step, a quadratic-linear approximation of the recursive Bellman equation (5.14), around the initial trajectory of the iteration, is solved, leading to an improved trajectory. The improved trajectory is then used as initial trajectory in the next iteration, and the procedure continues until convergence to the solution of the original optimal control problem is achieved. The method was developed for deterministic optimal control problems; and it is not applicable to general stochastic problems, which do not feature solution trajectories (Bertsekas, 2012). However, as explained earlier, the specific stochastic GLOSA problem admits feasible solution trajectories for specific initial states, hence DDP may be applied.

5.5.1 General Constrained DDP

The standard DDP method (Jacobson, Mayne, 1970) does not account for constraints, so, for the needs of the problem presented in this work, extensions of DDP, proposed by Murray,

Yakowitz (1979) for linear constraints and by Yakowitz (1986) for nonlinear constraints, is utilized.

Consider a discrete-time optimal control problem with states \mathbf{x} and controls \mathbf{a} with the recursive Bellman equation $V(\mathbf{x}(k), k) = \min\{\phi(\mathbf{x}, \mathbf{a}, k) + V(\mathbf{f}(\mathbf{x}, \mathbf{a}, k), k + 1)\}$, for $k = K - 1, \dots, 0$, where V is the optimal cost function, ϕ is the objective function of the optimal control problem, \mathbf{f} is the right-hand side of the state equation, and the minimisation must be carried out for all feasible controls $\mathbf{a}(k)$ that satisfy any inequality constraints present. The DDP procedure undertakes, at each time step of each iteration, a quadratic approximation of the term to be minimized in the recursive Bellman equation, i.e., of $\phi(\mathbf{x}, \mathbf{a}, k) + V(\mathbf{f}(\mathbf{x}, \mathbf{a}, k), k + 1)$. The quadratic approximation is taken around nominal trajectories $(\bar{\mathbf{x}}(k), \bar{\mathbf{a}}(k))$, which are the initial trajectories of each iteration. Define $\delta\mathbf{x} = \mathbf{x} - \bar{\mathbf{x}}$ and $\delta\mathbf{a} = \mathbf{a} - \bar{\mathbf{a}}$. The aim is to find the optimal control law for $\delta\mathbf{a}$, which minimizes the quadratic approximation, subject to some inequality constraints. The procedure of the DDP algorithm for each iteration is described in brief as follows, see (Murray, Yakowitz, 1979; Yakowitz, 1986) for more details.

For $k = K - 1, \dots, 0$, the quadratic approximation $Q(\mathbf{x}, \mathbf{a}, k)$ of the term $\phi(\mathbf{x}, \mathbf{a}, k) + V(\mathbf{f}(\mathbf{x}, \mathbf{a}, k), k + 1)$ around a nominal point $(\bar{\mathbf{x}}(k), \bar{\mathbf{a}}(k))$, is required and can be expressed as follows (for simplicity, the indexes k are omitted)

$$Q(\mathbf{x}, \mathbf{a}) = \frac{1}{2} \delta\mathbf{x}^T \mathbf{A} \delta\mathbf{x} + \delta\mathbf{x}^T \mathbf{B} \delta\mathbf{a} + \frac{1}{2} \delta\mathbf{a}^T \mathbf{C} \delta\mathbf{a} + \mathbf{D}^T \delta\mathbf{a} + \mathbf{E}^T \delta\mathbf{x} \quad (5.18)$$

where the zero-order constant term is omitted, as it does not influence the minimisation; and $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}$ are matrices with appropriate dimensions and with elements that may depend on $(\bar{\mathbf{x}}(k), \bar{\mathbf{a}}(k))$.

Furthermore, we assume the general inequality constraints for the optimal control problem

$$\mathbf{c}(\mathbf{x}, \mathbf{a}) \leq \mathbf{0} \quad (5.19)$$

and we denote as $\tilde{\mathbf{c}}(\mathbf{x}, \mathbf{a})$ its linear approximation around the nominal trajectories $(\bar{\mathbf{x}}, \bar{\mathbf{a}})$.

For the solution of the resulting Quadratic Programming (QP) problem, Fletcher's active set algorithm is used (Fletcher, 1971). The critical issue here is that the minimization in the recursive Bellman equation should be executed in a parametrized way, i.e., deliver $\delta\mathbf{a}$ as a function of $\delta\mathbf{x}$, which is not compatible with a numerical solution procedure. Therefore, in the QP problem formulation, we replace \mathbf{x} with $\bar{\mathbf{x}}$ (i.e. we set $\delta\mathbf{x} = 0$) in (5.18) and $\tilde{\mathbf{c}}(\mathbf{x}, \mathbf{a})$, and

solve the following QP problem to obtain a numerical value for $\delta \mathbf{a}$

$$\begin{aligned} & \min Q(\bar{\mathbf{x}}, \bar{\mathbf{a}} + \delta \mathbf{a}) \\ & \text{s.t} \\ & \tilde{\mathbf{c}}(\bar{\mathbf{x}}, \bar{\mathbf{a}} + \delta \mathbf{a}) \leq \mathbf{0} \end{aligned} \quad (5.20)$$

After solving the QP problem, the index set of the active constraints, denoted as S , leads to the linear system of equations

$$\tilde{c}_i(\bar{\mathbf{x}}, \bar{\mathbf{a}} + \delta \mathbf{a}) = 0, \quad i \in S \quad (5.21)$$

and expressing (5.21) in matrix form, we have

$$\mathbf{U} \delta \mathbf{a} - \mathbf{W} = \mathbf{0} \quad (5.22)$$

where \mathbf{U} has rank equal to the number of indexes in S . This solution of the QP problem satisfies the necessary conditions of optimality for the Lagrangian (Fletcher, 1971)

$$L(\mathbf{a}, \boldsymbol{\lambda}) = \frac{1}{2} \delta \mathbf{a}^T \mathbf{C} \delta \mathbf{a} + \mathbf{D}^T \delta \mathbf{a} + \boldsymbol{\lambda}^T (\mathbf{U}^T \delta \mathbf{a} - \mathbf{W}) \quad (5.23)$$

where $\boldsymbol{\lambda}$ is the Lagrange multiplier vector. These necessary conditions may be obtained by differentiating L with respect to $\delta \mathbf{a}$ and $\boldsymbol{\lambda}$, which yields the following linear system of equations

$$\begin{bmatrix} \mathbf{C} & \mathbf{U} \\ \mathbf{U}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \delta \mathbf{a} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\mathbf{D} \\ \mathbf{W} \end{bmatrix} \quad (5.24)$$

In order to re-introduce $\delta \mathbf{x}$, i.e. the variability of state variables, we consider that the constraints that are active while we assumed $\mathbf{x} = \bar{\mathbf{x}}$, remain active for other \mathbf{x} , i.e.

$$\tilde{c}_i(\bar{\mathbf{x}} + \delta \mathbf{x}, \bar{\mathbf{a}} + \delta \mathbf{a}) = 0, \quad i \in S \quad (5.25)$$

which in matrix form reads

$$\mathbf{U} \delta \mathbf{a} - \mathbf{W} - \mathbf{X} \delta \mathbf{x} = \mathbf{0} \quad (5.26)$$

Then, we may derive a control law, i.e. $\delta \mathbf{a}$ as a function of $\delta \mathbf{x}$ by solving analytically the accordingly extended version of (5.24), i.e.

$$\begin{bmatrix} \mathbf{C} & \mathbf{U} \\ \mathbf{U}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \delta \mathbf{a} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\mathbf{D} - \mathbf{B} \delta \mathbf{x} \\ \mathbf{W} + \mathbf{X} \delta \mathbf{x} \end{bmatrix} \quad (5.27)$$

which yields the following linear control law with readily computed vector $\boldsymbol{\alpha}$ and matrix $\boldsymbol{\beta}$

$$\delta \mathbf{a}(\mathbf{x}) = \boldsymbol{\alpha} + \boldsymbol{\beta} \delta \mathbf{x} \quad (5.28)$$

This minimisation outcome is substituted in the quadratic approximation Q to obtain, from the Bellman equation, the approximate optimal cost function

$$V(\mathbf{x}(k), k) = Q(\mathbf{x}(k), \boldsymbol{\alpha}(k) + \boldsymbol{\beta}(k) \delta \mathbf{x}(k), k) \quad (5.29)$$

which must be stored for use at the next time step $k - 1$ within the same iteration. Note that this approximate cost function is quadratic by construction, which facilitates the quadratic approximation in the Bellman equation at the next time step $k - 1$.

Given $\delta \mathbf{a}(\delta \mathbf{x}, k)$, from (5.28), the nominal trajectories for the next iteration are calculated as follows for $k = 0, \dots, K - 1$

$$\mathbf{a}(k) = \bar{\mathbf{a}}(k) + \varepsilon [\boldsymbol{\alpha}(k) + \boldsymbol{\beta}(k) \delta \mathbf{x}(k)] \quad (5.30)$$

$$\mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{a}(k), k) \quad (5.31)$$

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (5.32)$$

where $0 < \varepsilon \leq 1$ is used to ensure the convergence of the algorithm. Specifically, ε values smaller than 1 may facilitate convergence if needed; but may decrease the convergence rate if selected too small. The algorithm terminates whenever the following convergence test is satisfied

$$\left(\sum_{k=0}^{K-1} [(\mathbf{a}(k) - \bar{\mathbf{a}}(k))^2] \right)^{1/2} < \varepsilon_1 \quad (5.33)$$

where ε_1 is a small positive number.

In contrast to the SDP and DDDP algorithms, DDP does not apply any discrete DP procedures, hence no discretization of the state and control domains is necessary. As a result, the required computation time is independent of any discretization intervals, and the obtained solution trajectories are value-continuous, hence potentially better than those obtained by the other algorithms, even when using fine discretization.

5.5.2 Constrained DDP for the GLOSA Problem

To apply the DDP algorithm to the GLOSA stochastic optimal control problem described in Section 5.3.2, the quadratic approximation of the right-hand side of (5.14) around a nominal trajectory $(\bar{\mathbf{x}}(k), \bar{\mathbf{a}}(k))$ is first required. Note that only the J_{DG}^* term needs to be approximated, as the rest terms are already quadratic. For the approximation, the LOESS (locally estimated

scatterplot smoothing) method was utilized (Cleveland et al., 2017). Specifically, a number of (x, v) points are specified around the nominal trajectory at each iteration, and the corresponding values of $J_{DG}^*(\mathbf{x}(k), kT)$ are computed. It was found that some 4 such points per time step are sufficient for a good fit. Then, LOESS fits the following smooth function to the data set

$$p(x, v) = p_1 \delta x^2 + p_2 \delta v^2 + p_3 \delta x \delta v + p_4 \delta x + p_5 \delta v + p_6 \quad (5.34)$$

where p_1, \dots, p_6 are the coefficients of fitting. The resulting fitted function (5.34), along with the other quadratic terms of (5.14), can be easily expressed in the form of (5.18). It was verified that the resulting quadratic function (5.18) is always convex, something that was expected from the physical background of the problem.

The nominal trajectory of the first iteration is taken to be the optimal solution of the deterministic GLOSA problem, assuming the "pessimistic" case where the traffic light switches from red to green at the latest possible time, i.e., $k_1 = k_{max}$.

For the state constraints (5.6), we may replace the state equations (5.4), (5.5) therein, hence, for the admissible region of the SDP (5.6) and (5.7), we have the equivalent constraints

$$\begin{aligned} \frac{2(x_{min} - x(k) - v(k)T)}{T^2} &\leq a(k) \leq \frac{2(x_{max} - x(k) - v(k)T)}{T^2} \\ \frac{v_{min} - v(k)}{T} &\leq a(k) \leq \frac{v_{max} - v(k)}{T} \\ a_{min} &\leq a(k) \leq a_{max} \end{aligned} \quad (5.35)$$

which can be written in terms of δx and δa as follows

$$\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} \delta a - \begin{bmatrix} \frac{2(x_{max} - \bar{x}(k+1))}{T^2} \\ -2(x_{min} - \bar{x}(k+1)) \\ \frac{T^2}{(v_{max} - \bar{v}(k+1))} \\ \frac{T}{-(v_{min} - \bar{v}(k+1))} \\ \frac{T}{(a_{max} - \bar{a}(k))} \\ -(a_{min} - \bar{a}(k)) \end{bmatrix} - \delta \mathbf{x} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (5.36)$$

Having the quadratic approximation of the recursive Bellman equation and the linear inequality constraints (5.35), the resulting QP is as described in the previous subsection. In fact, the resulting QP problem for the stochastic GLOSA application is simpler than the general case of Section 5.5.1. After setting $\delta \mathbf{x} = 0$ in (5.36), we have only one decision

variable, δa , with constant upper and lower bounds. Thus, for the solution of the QP problem, it suffices to first compute analytically the value of δa that minimises the quadratic function. If this value does not violate any of the upper or lower bounds, then the QP solution is given by that value; else, the QP solution equals the activated bound. All further procedures are as explained in Section 5.5.1.

5.6 Demonstration Results and Comparison

In this section, the performance of the proposed approaches is tested, demonstrated and compared. Specifically, the DDDP and DDP algorithms are applied to realistic scenarios and evaluated regarding the quality of their solution compared to the standard SDP, as well as the computational time needed for each algorithm to converge to the optimal solution.

5.6.1 Scenarios Setup

For the evaluation, several scenarios have been considered in various situations, such as cases where the vehicle needs to accelerate or decelerate before or after the traffic signal; or cases where the vehicle is forced to fully stop and wait until the traffic light switches from red to green. For this paper, three scenarios have been chosen for demonstration with full illustration of state and control trajectories, according to Table 1. For consistency, the chosen scenarios are the same as in previous related works (Typaldos et al., 2020a, 2021). All scenarios concern mixed acceleration-deceleration cases. However, in Scenario 1, the vehicle starts with low initial speed and needs to mostly accelerate in order to optimally pass the traffic signal. On the other hand, in Scenarios 2 and 3, the vehicle starts at higher speed, hence it needs to mostly decelerate to avoid a full stop at the traffic signal. Compared to Scenarios 1 and 2, in Scenario 3 the vehicle starts closer to the traffic light, which leads to a full stop for few time steps before the traffic signal switches from red to green.

Table 5.1: Investigated scenarios.

	x_0 (m)	v_0 (m/s)	x_e (m)	v_e (m/s)	x_1 (m)
Scenario 1	0	5	220	11	150
Scenario 2	0	11	220	11	150
Scenario 3	50	11	220	11	150

For all evaluated scenarios, the weight parameter w in (5.3) is equal to 0.1. The bounds for the states and control are set to $[x_{min}, x_{max}] = [0, 150]$ m, $[v_{min}, v_{max}] = [0, 16]$ m/s and $[a_{min}, a_{max}] = [-3, 3]$ m/s², respectively. The time step T is 1 s and the switching time window

for the traffic signal is $[k_{min}, k_{max}] = [10, 30]$ s with uniform a-priori probability distribution. Regarding DDDP, for the initial discretization, $\Delta a = \Delta v = 0.5$ is used, which leads to $\Delta x = 0.25$ m according to (5.15); and the initial corridor width in (5.17) is set $\mathbf{C} = [20, 4]$, i.e., we have $C_x = 5 \cdot C_v$. The choice of the initial discretization and corridor width value will be justified later in this section. For the DDP algorithm, parameters ε in (5.30) and ε_1 in (5.33) are set equal to 1 and 10^{-4} , respectively.

5.6.2 DDDP Algorithm Results

Fig. 5.2 (Scenario 1) and Figs. 5.6 and 5.7 (for Scenarios 2 and 3, respectively, see Appendix A) display the optimal state (speed) and control (acceleration) trajectories over each iteration of the DDDP algorithm. Note that, position trajectories are not included, as the differences in those trajectories, at each iteration, are small and barely visible. In each iteration, we consider a corridor $\Delta_C^{(l)} = [-\mathbf{C} \cdot \Delta a^{(l)}, \mathbf{C} \cdot \Delta a^{(l)}]$ around the respective received state trajectories, which cannot of course extend out of the full state bounds.

In all Figs 5.2, 5.6 and 5.7, the dashed blue lines represent, for each iteration, the received trajectory, the solid orange lines represent the optimal trajectories derived, and the red dashed lines reflect the corresponding corridor bounds. For Scenario 1 (Fig. 5.2), it can be observed that, starting with the initial chosen discretization, the first DDDP iteration improves the initial trajectory, leading to a better solution, which is optimal within the considered sub-domain. In the second iteration, no further improvement can be achieved within the new sub-domain bounds, which means that, with the current discretization values, the best achievable solution has been reached. By reducing the discretization at the 3rd iteration, a reduction in the corridor width is observed, but the number of discrete points remains the same. This reduction enables further improvement of the solution, which remains the same in iteration 4; hence the discretisation is further reduced, leading to further improvement in iteration 5; and the procedure continues with iteration 6, where no further improvement is reached and the algorithm terminates, as the discretisation has reached the minimum required value. Similar behaviour is observed for Scenarios 2 and 3, where 10 iterations are required for convergence, and the respective Figs. 5.6 and 5.7 display a subset thereof. Table 5.2 contains the values of the objective function for each iteration of the DDDP algorithm for all scenarios, assuming $\Delta a = 0.5$ and $\mathbf{C} = [20, 4]$, where the choice of these values will be explained in the following.

The DDDP has been evaluated for different corridor C_v and initial discretization Δa values, for the three investigated scenarios (see Table 5.4 in Appendix A). It can be seen that, for all presented initial discretization values, the obtained optimal solution at convergence reaches the same globally optimal solution, if the corridor width is sufficiently large, specifically for $C_v \geq 4$ (or lower, in some cases). This is because, for very small values of C_v , DDDP does not have sufficient flexibility to improve the solution at each iteration due to the extremely limited state

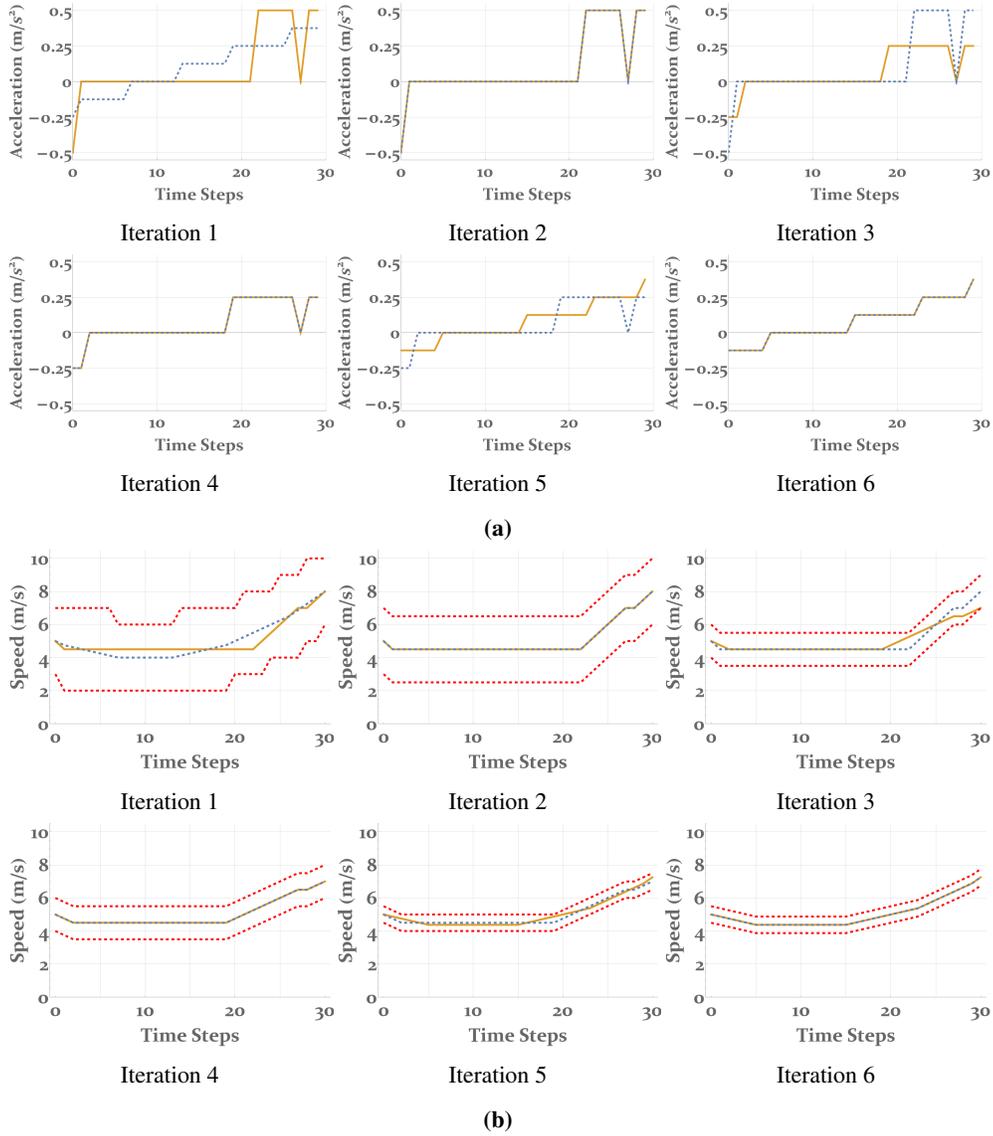


Figure 5.2: Received (blue dashed line) and optimal (orange line) (a) acceleration and (b) speed trajectories of DDDP algorithm in each iteration for Scenario 1.

Table 5.2: Optimal cost evolution and discretization change in each iteration of DDDP algorithm.

Iter.	Scenario 1			Scenario 2			Scenario 3		
	$\Delta a = \Delta v$	Δx	Cost	$\Delta a = \Delta v$	Δx	Cost	$\Delta a = \Delta v$	Δx	Cost
1	0.5	0.25	1.357	0.5	0.25	4.311	0.5	0.25	6.6218
2	0.5	0.25	1.357	0.5	0.25	4.231	0.5	0.25	6.6218
3	0.25	0.125	1.223	0.5	0.25	4.231	0.25	0.125	6.4532
4	0.25	0.125	1.223	0.25	0.125	4.012	0.25	0.125	6.4346
5	0.125	0.0625	1.175	0.25	0.125	3.988	0.25	0.125	6.4208
6	0.125	0.0625	1.175	0.25	0.125	3.958	0.25	0.125	6.4091
7				0.25	0.125	3.940	0.25	0.125	6.4091
8				0.25	0.125	3.940	0.125	0.0625	6.3583
9				0.125	0.0625	3.906	0.125	0.0625	6.3582
10				0.125	0.0625	3.906	0.125	0.0625	6.3582

space, something that may lead to a local optimum of the overall problem at convergence. At the same time, a reduction of the number of DDDP iterations is obtained for larger corridors. This behaviour is expected, as the bigger corridors, i.e., bigger admissible sub-regions, lead to potentially better solutions at each iteration, and hence to fewer iterations to converge to the optimal solution. On the other hand, despite the decrease on the number of iterations, the overall computation time may increase due to the higher computation time required at each iteration, which in turn, is due to more feasible discrete state points included in each iteration's sub-region. Based on these observations, the selection of the values $\mathbf{C} = [20, 4]$ and $\Delta a = 0.5$ seems to be a reasonable choice.

The accuracy of the DDDP algorithm compared to the full one-shot SDP solution may be assessed by contrasting the optimal cost and the optimal states and control trajectories, obtained in the last respective DDDP iterations of the three scenarios, to the corresponding optimal trajectories derived from the full-range SDP with a discretisation of $\Delta a = 0.125$. As a matter of fact, DDDP manages, in all cases, to converge to the exact same optimal solution as the full-range SDP. The obtained optimal costs of both approaches, for the three scenarios, are 1.175, 3.906 and 6.3582, respectively, while the computation time differences are remarkable, as the one-shot SDP needs approximately 613 s to obtain the solution (same for all scenarios), while DDDP needs only 0.69, 1.24 and 1.12 s for the respective three scenarios. More importantly, this big reduction in computation time enables the DDDP algorithm to be executable in real time, even in an MPC mode, on the vehicle side, similarly to the deterministic GLOSA.

5.6.3 DDP Algorithm Results

Fig. 5.3 (Scenario 1) and Figs. 5.8 and 5.9 (for Scenarios 2 and 3, respectively, see Appendix A) illustrate the evolution of the state (speed) and control (acceleration) trajectories produced by the DDP algorithm at each iteration. In all figures, blue dashed lines represent the initial trajectories, and solid orange lines the derived trajectories of each iteration. From the figures, it can be seen that the DDP algorithm, starting from the pessimistic deterministic GLOSA solution, converges to the overall optimal solution. It can also be observed that, at each iteration, starting from the initial (received) trajectory, the algorithm results in an improved one, which leads to an improved solution. The DDP algorithm is seen to converge very fast to the optimal solution, with the improvement being more profound at the first iterations, as the change between the nominal and the produced trajectories are more noticeable. On the other hand, in the last iterations, the improvement is smaller, as the algorithm is already very close to the optimal solution. As mentioned earlier, all intermediate DDP solutions at each iteration are continuous due to the nature of DDP, which is operating on a continuous-value basis for all involved variables.

Fig. 5.4 demonstrates the accuracy of the DDP solutions compared to the one-shot discrete SDP solutions for the three scenarios. In this figure, the optimal state (speed) and control (acceleration) trajectories obtained from DDP algorithm at convergence are contrasted to those obtained from the full-range SDP algorithm with discretization $\Delta a = 0.125 \text{ m/s}^2$. The two solutions, in all scenarios, are quite close to each other, with a slight advantage for the DDP solution, which is not bound to a fine, but still visible value-discretization of the state and control, as it is the case for the one-shot discrete SDP algorithm.

5.6.4 Summarized Results

Table 5.3 contains the summarized information of the number of iterations until convergence, CPU time and the optimal obtained cost of the one-shot SDP, the iterative DDDP and the iterative, but non-discrete DDP algorithms for Scenarios 1, 2 and 3. Based on the results reported in the table, it can be verified that DDP's optimal solution is slightly better compared to the other two approaches. However, the most striking difference of DDP is the computational time needed to solve the problem, which is extremely small.

In order to explore the robustness of the DDDP and DDP algorithms and to further emphasize their superiority compared to the one-shot (S)DP, in terms of computation times, a more excessive analysis was conducted and is presented in Fig. 5.5. Specifically, Fig. 5.5 illustrates the required CPU times until convergence for both proposed algorithms for a plethora of different initial conditions, i.e., initial speeds and positions (distance from the traffic signal) of a vehicle. From both Fig. 5.5a and Fig. 5.5b it can be seen that, for all

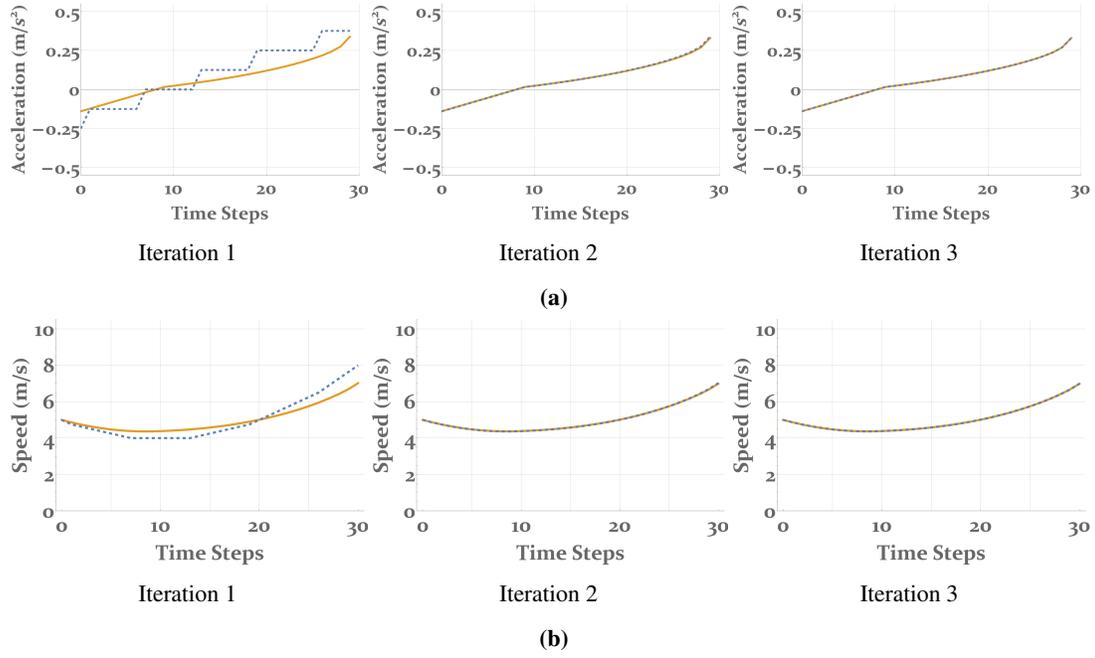


Figure 5.3: Received (blue dashed line) and computed (orange line) trajectories of (a) acceleration and (b) speed at each iteration of DDP algorithm for Scenario 1.

Table 5.3: Comparison of iterations, CPU times until convergence and obtained optimal cost of SDP, DDDP and DDP

	Scenario 1			Scenario 2			Scenario 3		
	Iter.	CPU Times (s)	Cost	Iter.	CPU Times (s)	Cost	Iter.	CPU Times (s)	Cost
One-shot SDP	-	631	1.175	-	631	3.906	-	631	6.358
DDDP	6	0.692	1.175	10	1.246	3.906	10	1.128	6.358
DDP	3	0.0005	1.162	4	0.0007	3.892	4	0.0007	6.353

evaluated initial conditions, the CPU times of both DDDP and DDP are similarly low as for the above three scenarios, with average times of 1.4 s and 0.001 s, respectively, the corresponding maximum and minimum times being 6.3 s and 0.2 s for DDDP and 0.006 s and

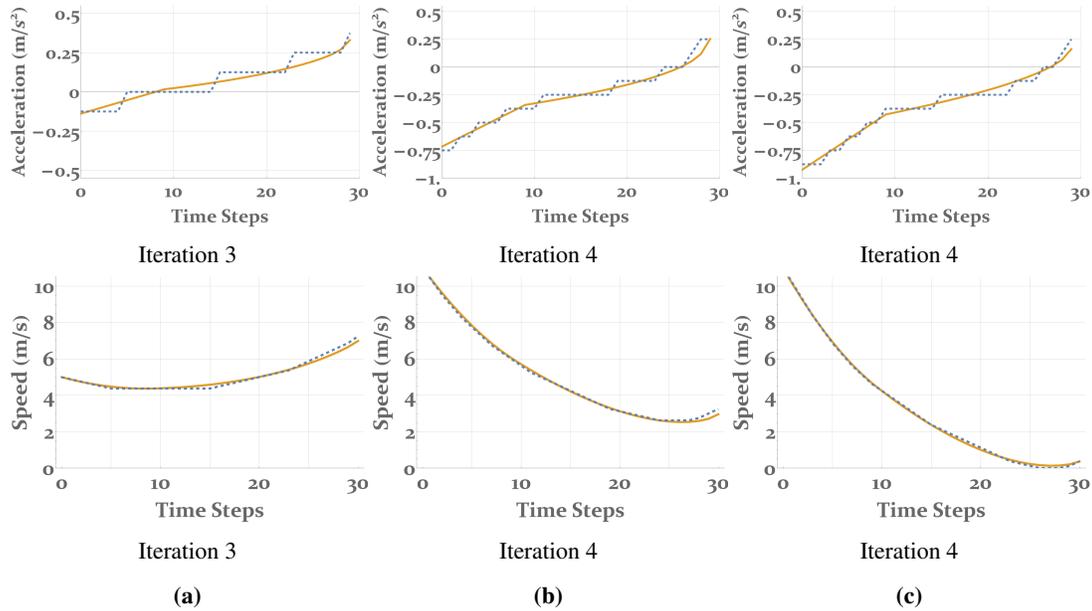


Figure 5.4: Optimal acceleration and speed trajectories (orange lines) of DDP, compared with the corresponding optimal trajectories of the one-shot SDP (blue dashed lines) of (a) Scenario 1, (b) Scenario 2 and (c) Scenario 3.

0.0005 s for DDP, respectively. These findings lead to the conclusion that, even though DDDP reduces the CPU time significantly compared to the one-shot SDP, the DDP algorithm achieves almost instantaneous solutions, which enables the DDP approach to be used, within an MPC framework, in the vehicle’s on-board computer.

5.7 Conclusions

In recent works (Typaldos et al., 2020a, 2021), a stochastic GLOSA methodology was developed, by optimizing, using SDP and DDDP techniques, respectively, the vehicle kinematic trajectories subject to the intermediate stochastic traffic signal switching constraint and with a fixed final state. As an extension to these works, an iterative Differential Dynamic Programming (DDP) algorithm was employed in this paper. Within each iteration, DDP solves, for each time-step separately, a quadratic-linear approximation of the recursive Bellman equation. Demonstration and comparison results indicate that the DDP algorithm strongly outperforms both the full-range SDP and the DDDP algorithms in terms of

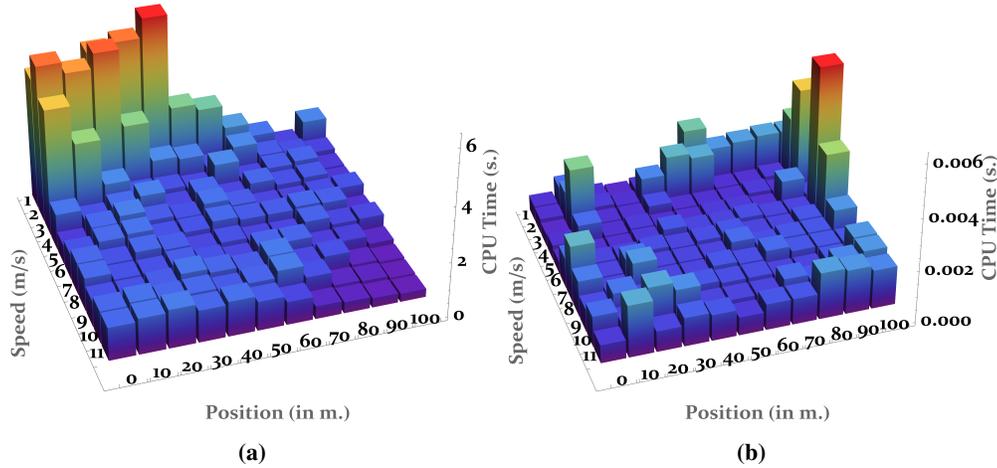


Figure 5.5: Computation times until convergence of: (a) DDDP and (b) DDP algorithms for different initial conditions.

computation time; and even delivers a slightly better optimal solutions due to lack of discretisation. This enables the DDP algorithm to be readily executable within a model predictive control framework in real time on-board approaching vehicles.

Current and future work is focused on:

- Generalization of the current GLOSA problem by considering uncertain switching times for both green and red phases.
- Testing of the approach performance in presence of multiple (equipped and not equipped) vehicles, while traversing multiple subsequent intersections.

Acknowledgments

The research leading to these results has received funding form the European Research Council under the European Union’s Horizon 2020 Research and Innovation programme/ERC Grant Agreement n. [833915], project TrafficFluid.

Appendix A

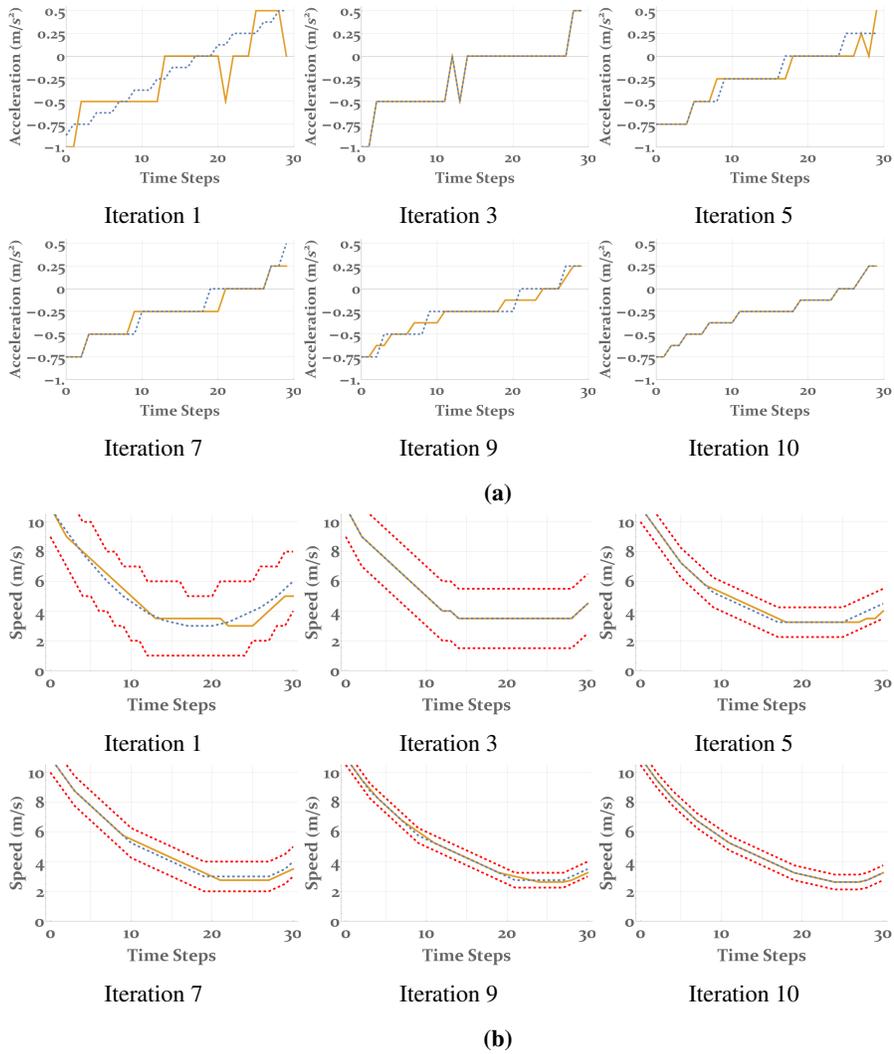


Figure 5.6: Received (blue dashed line) and optimal (orange line) (a) acceleration and (b) speed trajectories of DDDP algorithm in each iteration for Scenario 2.

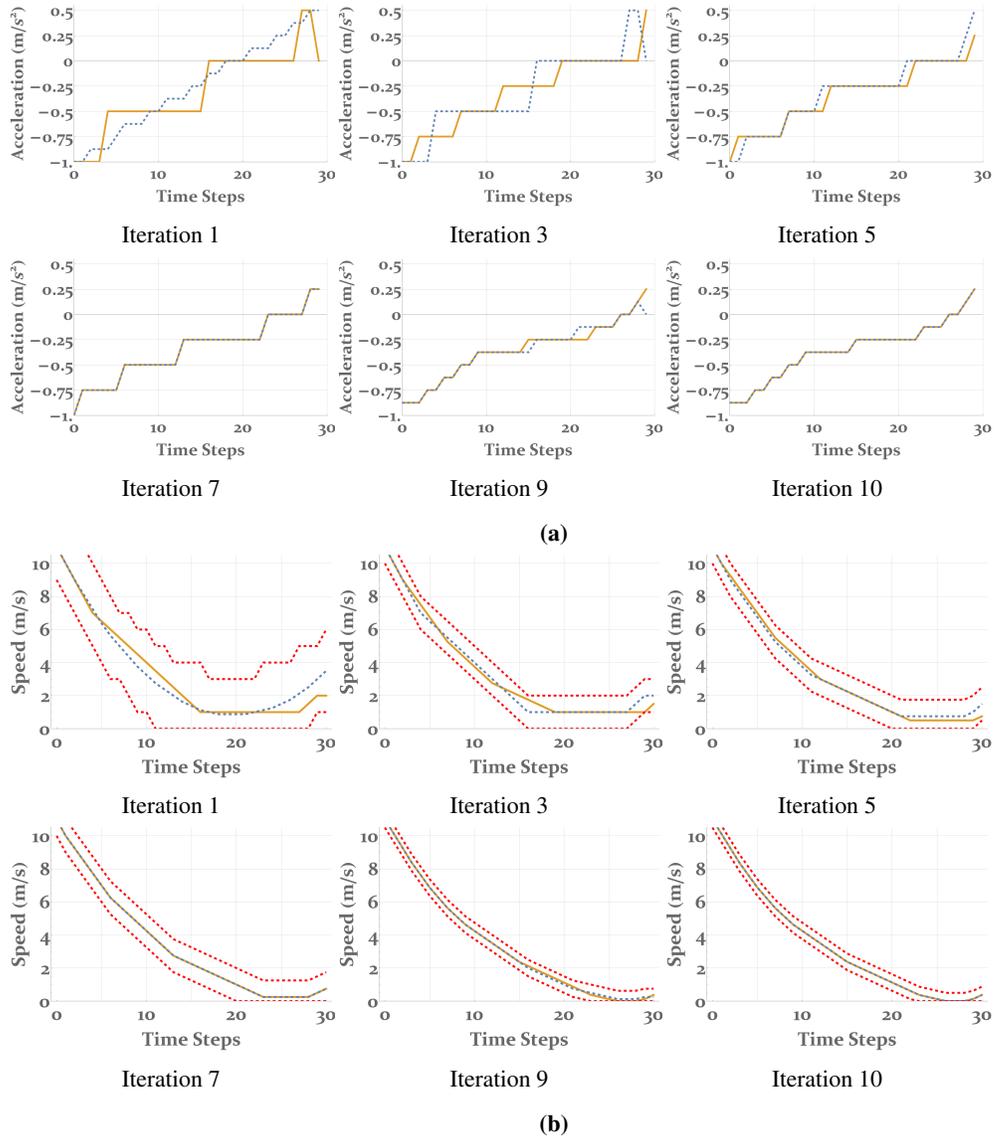


Figure 5.7: Received (blue dashed line) and optimal (orange line) (a) acceleration and (b) speed trajectories of DDDP algorithm in each iteration for Scenario 3.

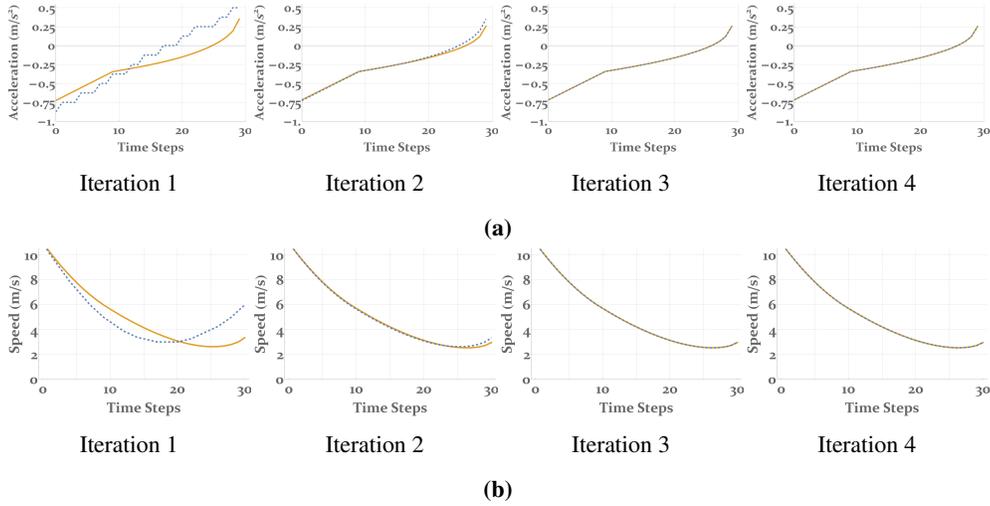


Figure 5.8: Received (blue dashed line) and computed (orange line) trajectories of (a) acceleration and (b) speed at each iteration of DDP algorithm for Scenario 2.

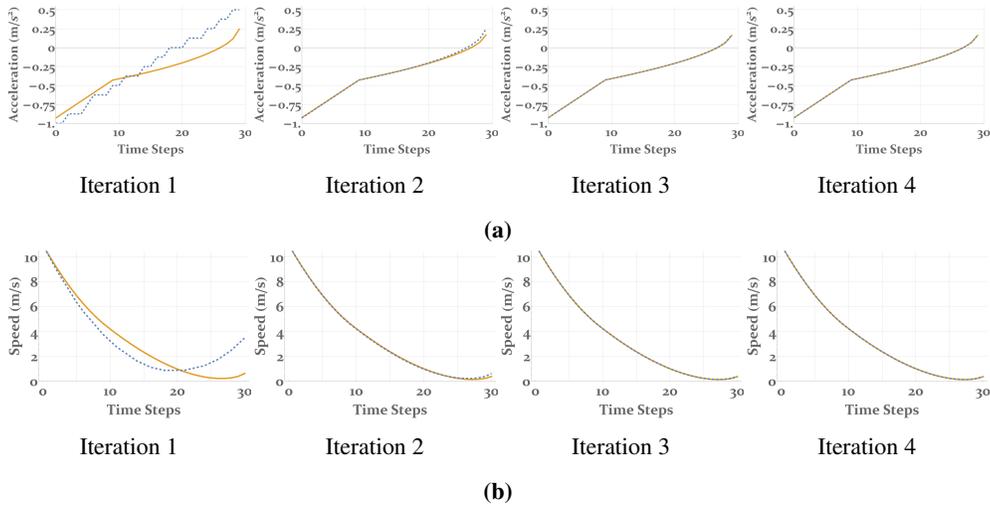


Figure 5.9: Received (blue dashed line) and computed (orange line) trajectories of (a) acceleration and (b) speed at each iteration of DDP algorithm for Scenario 3.

Table 5.4: Performance of DDDP algorithm in terms of CPU-time and optimal cost for different initial values of \mathbf{C} and Δa .

Scenario 1												
$\Delta a = 1.0$				$\Delta a = 0.5$			$\Delta a = 0.25$			$\Delta a = 0.125$		
C_v	Iter.	CPU Times (s)	Cost	Iter.	CPU Times (s)	Cost	Iter.	CPU Times (s)	Cost	Iter.	CPU Times (s)	Cost
2	19	0.648	1.199	20	0.612	1.193	10	0.326	1.193	5	0.280	1.193
3	19	1.296	1.177	8	0.552	1.175	6	0.469	1.179	5	0.443	1.179
4	15	1.655	1.175	6	0.692	1.175	7	0.867	1.175	5	0.762	1.175
5	12	1.839	1.175	6	1.052	1.175	6	1.111	1.175	4	0.885	1.175
6	10	2.298	1.175	6	1.451	1.175	5	1.291	1.175	3	0.967	1.175
Scenario 2												
$\Delta a = 1.0$				$\Delta a = 0.5$			$\Delta a = 0.25$			$\Delta a = 0.125$		
C_v	Iter.	CPU Times (s)	Cost	Iter.	CPU Times (s)	Cost	Iter.	CPU Times (s)	Cost	Iter.	CPU Times (s)	Cost
2	29	1.212	3.909	18	0.790	3.909	16	0.729	3.909	21	1.014	3.909
3	22	1.656	3.906	14	1.093	3.906	12	0.966	3.906	18	1.568	3.906
4	13	1.578	3.906	10	1.246	3.906	9	1.149	3.906	13	1.740	3.906
5	11	1.920	3.906	7	1.273	3.906	8	1.514	3.906	11	2.178	3.906
6	14	3.324	3.906	10	2.471	3.906	7	1.823	3.906	9	2.457	3.906
Scenario 3												
$\Delta a = 1.0$				$\Delta a = 0.5$			$\Delta a = 0.25$			$\Delta a = 0.125$		
C_v	Iter.	CPU Times (s)	Cost	Iter.	CPU Times (s)	Cost	Iter.	CPU Times (s)	Cost	Iter.	CPU Times (s)	Cost
2	28	1.152	6.3583	20	0.868	6.3583	18	0.815	6.3583	21	0.998	6.3644
3	21	1.543	6.3582	16	1.192	6.3582	14	1.098	6.3582	17	1.419	6.3639
4	14	1.506	6.3582	10	1.128	6.3582	10	1.179	6.3582	14	1.834	6.3582
5	12	1.777	6.3582	8	1.253	6.3582	8	1.400	6.3582	11	2.101	6.3582
6	11	2.050	6.3582	7	1.368	6.3582	7	1.553	6.3582	9	2.328	6.3582

Chapter 6

GLOSA System with Uncertain Green and Red Signal Phases

6.1 Abstract

A discrete-time stochastic optimal control problem was recently proposed to address the GLOSA (Green Light Optimal Speed Advisory) problem in cases where the next signal switching time is decided in real-time and is therefore uncertain in advance. However, there was an assumption that the traffic signal is initially red and turns to green, which means that only half traffic light cycle was considered. In this work, the aforementioned problem is extended considering a full traffic light cycle, consisting of four phases: a certain green phase, during which the vehicle can freely pass; an uncertain green phase, in which there is a probability that the traffic light will extend its duration or turn to red at any time; a certain red phase during which the vehicle cannot pass; and an uncertain red phase, in which there is a probability that the red signal may be extended or turn to green at any time. It is demonstrated, based on preliminary results, that the proposed SDP (Stochastic Dynamic Programming) approach achieves better average performance, in terms of fuel consumption, compared to the IDM (Intelligent Driver Model), which emulates human driving behavior.

6.2 Introduction

A common dilemma for a vehicle approaching a traffic light, is whether it should maintain its speed or should accelerate to cross (if the current signal is green) or decelerate to avoid stopping (if the current signal is red). To this end, many systems have been developed which aim at guiding the driver (or the automated vehicles) by giving speed advise which ensures that the vehicle will cross the traffic signal during the green phase and with minimum fuel

consumption and emissions. Such systems are often referred to as Green Light Optimal Speed Advisory (GLOSA) systems (Stahlmann et al., 2016).

In the case of fixed signals and hence prior knowledge of the next switching time, the signal controller may broadcast a corresponding message to approaching vehicles. Under these conditions, the problem of how to optimize the approach to the traffic signals has been addressed in different ways (Katsaros et al., 2011; Lawitzky et al., 2013; Sanchez et al., 2006). The situation becomes more complicated when real-time signals with very short (e.g., second-by-second) control update periods are present, in which case exact prior knowledge of the next switching time is not available. In this case, the best available knowledge can be presented as an estimate (Koukoumidis et al., 2011) or as a probabilistic distribution for the next switching time within a future time-window; such a distribution may be obtained by use of statistics from previous signal operation (Lawitzky et al., 2013; Mahler, Vahidi, 2012; Sun et al., 2020).

In (Typaldos et al., 2020a), the problem of producing fuel-optimal vehicle trajectories for a vehicle approaching a traffic signal, for both cases of known and stochastic switching times, was considered. For the first case, the problem was formulated as an optimal control problem and was solved analytically via PMP (Pontryagin's Maximum Principle). Subsequently, the case of stochastic switching time with known probability distribution was also addressed, and the problem was cast in the format of a stochastic optimal control problem, which was solved numerically using SDP (Stochastic Dynamic Programming). However, there was an assumption that the traffic signal is initially red and turns to green, which means that only the half traffic light cycle was considered.

The present work delivers an extension of the problem by considering a full traffic light cycle. Specifically, the signal's cycle consists of four phases, i.e., a certain-switching green phase, an uncertain-switching green phase, a certain-switching red phase, and an uncertain-switching red phase (see Fig. 6.1). As mentioned, in our previous work, the vehicle was assumed to appear during the last two phases (certain red and uncertain red), where the certain red phase has a fixed and known switching time, followed by the uncertain red phase, in which there is at any time a probability that red will be extended, or that it will turn to green. The addition of the two green phases completes the handling of the GLOSA problem, as the vehicle may now appear at any of the four phases, with the certain-green phase having a fixed and known switching time, followed by the uncertain green phase, in which there is again at any time a probability that the traffic light may extend or turn red. Preliminary results demonstrate that the proposed SDP approach achieves better performance, in terms of fuel consumption and passenger comfort, compared to the base case of IDM (Treiber, Kesting, 2013), which emulates the driving behavior of a manually driven vehicle.

6.3 Problem Formulation and Solution

The proposed stochastic GLOSA approach aims at guiding a vehicle, starting from an initial state \mathbf{x}_0 (comprising initial vehicle position and speed), to cross a traffic signal, located at position x_1 , at green; and reach a fixed final state \mathbf{x}_e , within a free (but penalized) time horizon t_e (Fig. 6.1). The final state \mathbf{x}_e comprises a specified position x_e downstream of the traffic signal and a specified, reasonably high speed v_e . Note that the SDP algorithm solves the problem for vehicle positions up to the traffic signal position, i.e., for $x_0 \leq x_1$; however, the use of a deterministic GLOSA solution is included in the problem formulation and enables the vehicle to "escape" to the final state (Typaldos et al., 2020a). Note also that the solution of the problem via SDP delivers an optimal feedback law, i.e., for every admissible state $\mathbf{x}(k)$, we get the corresponding optimal control (acceleration) $a(k)$.

For the stochastic optimal control problem, the vehicle kinematics in discrete-time, with time step T , are described as follows:

$$x(k+1) = x(k) + v(k)T + \frac{1}{2}a(k)T^2 \quad (6.1)$$

$$v(k+1) = v(k) + a(k)T \quad (6.2)$$

where $x(k), v(k)$ correspond to the vehicle position and speed at discrete times $k = 0, 1, \dots$ (where $kT = t$), while the control variable $a(k)$ is the acceleration that remains constant over each time period k . The state and control variables are bounded within the following admissible regions

$$\mathbf{x}(k) \in \mathbf{X} = [\mathbf{x}_{\min}, \mathbf{x}_{\max}] \quad (6.3)$$

$$a(k) \in U = [a_{\min}, a_{\max}] \quad (6.4)$$

with $\mathbf{x}_{\min}, \mathbf{x}_{\max}$ and a_{\min}, a_{\max} the lower and upper bounds of the states and acceleration, respectively. We consider the upper bound of the position, x_{\max} , to be the traffic light position x_1 .

A full signal cycle comprises four phases (see Fig. 6.1): certain-switching green, lasting $[0, k_{\min}^G - 1]$; uncertain-switching green, starting at k_{\min}^G and lasting at most until $k_{\max}^G - 1$; certain-switching red, lasting from its (uncertain) initial time until $k_{\min}^R - 1$; and uncertain-switching red, starting at k_{\min}^R and lasting at most until k_{\max}^R . We consider the case of fixed k_{\min}^G and k_{\min}^R , whereby any uncertain green portion that was not used, due to green-red switching at $k < k_{\max}^G$, is added to the certain-red phase, hence the uncertain-red window $[k_{\min}^R, k_{\max}^R]$ does not change due to "early" green-red switching; and the same applies when we have an "early" red-green switching. The case where the certain-red or certain-green time

durations are fixed, and hence the cycle duration reduces in case of "early" green-red or red-green switchings, may be treated similarly.

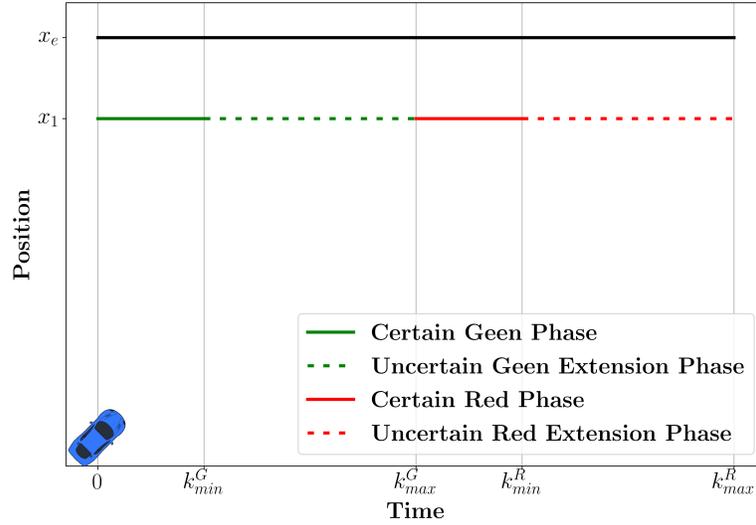


Figure 6.1: Traffic light phases.

A vehicle may appear at any time during a signal cycle, at an admissible initial state \mathbf{x}_0 , and be guided optimally to the final state \mathbf{x}_e . The green-red or red-green switching times are not known beforehand, but, when they actually occur, this is communicated to the approaching vehicles.

The total time horizon $[0, k_{max}^R]$ is subdivided in four parts as follows:

- Part 1: $[0, k_{min}^G - 1]$ (certain green)
- Part 2: $[k_{min}^G, k_{max}^G - 1]$ (uncertain green extension)
- Part 3: $[k_{max}^G, k_{min}^R - 1]$ (certain red)
- Part 4: $[k_{min}^R, k_{max}^R]$ (uncertain red extension).

To formally address the uncertainty in Parts 2 and 4, i.e., the uncertain green and uncertain red phases, we consider binary stochastic variables $z^G(k)$ and $z^R(k)$, respectively. The binary variables are equal to 0 if the traffic light switches to red or green, respectively, at time $k + 1$; or 1 else. We introduce virtual state variables $x^G(k)$ and $x^R(k)$, with initial values 1 and 0,

respectively, and respective state equations

$$x^G(k+1) = x^G(k)z^G(k) \quad (6.5)$$

$$x^R(k+1) = \begin{cases} 1 - z^G(k) & \text{if } x^R(k) = 0 \\ x^R(k)z^R(k) & \text{else} \end{cases} \quad (6.6)$$

hence

$$x^G(k) = \begin{cases} 1 & \text{if the green light has not yet switched until time } k-1 \\ 0 & \text{if switching occurred at time } k \text{ or earlier} \end{cases} \quad (6.7)$$

$$x^R(k) = \begin{cases} 0 & \text{if the green light has not yet switched until time } k-1 \\ 1 & \text{if the green light has switched, but the red light has} \\ & \text{not yet switched until time } k-1 \\ 0 & \text{if red - green switching occurred at time } k \text{ or earlier} \end{cases} \quad (6.8)$$

The virtual states $x^G(k), x^R(k)$ are assumed measurable, which means that the system knows, at each time kT , if switching has taken place or not within the last time-period $[(k-1)T, kT]$. Note in particular for Part 2 (green extension) that the vehicle is allowed to cross the traffic light position during any time period $[(k-1)T, kT]$, during which the switching may occur, since the vehicle decided its last acceleration at time $(k-1)T$, i.e., before the switching occurred. In other words, we may have $x(k) > x_{max}$ if $z^G(k-1) = 0$. This convention is not expected to jeopardise traffic safety, as time steps are short and accelerations and speeds are bounded.

The stochastic variables $z^G(k)$ and $z^R(k)$ are independent of their previous values and take values according to time-dependent probability distributions $p^G(z|k)$ and $p^R(z|k)$ respectively. Based on the statistics of previous signal switching activity, availability of a-priori discrete probability distributions $P^G(k), k_{min}^G \leq k \leq k_{max}^G$ and $P^R(k), k_{min}^R \leq k \leq k_{max}^R$, is assumed, with the involved probabilities summing up to 1. Based on these probabilities, we can calculate, using crop-and-scale, the required probabilities $p^G(z|k)$ and $p^R(z|k)$ for the stochastic variables $z^G(k)$ and $z^R(k)$ (Typaldos et al., 2020a).

The cost criterion of the stochastic problem is the same as in the deterministic GLOSA problem in (Typaldos et al., 2020a). However, in the stochastic case, the exact value of the criterion depends on the stochastic variables' realizations, and therefore we consider minimization of the expected value

$$J = E \left\{ w \cdot t_e + \frac{1}{2} \int_0^{t_e} a^2 dt \right\} \quad (6.9)$$

where the expectation refers to the stochastic variables $z^G(k), z^R(k), k = 0, \dots, k_{max} - 1$. Note that, when the vehicle crosses the traffic signal at state $\mathbf{x}(k)$, the problem instantly becomes a deterministic GLOSA problem, and the corresponding optimal cost-to-go is $J_{DG}^*(\mathbf{x}(k), k)$, as described in (Typaldos et al., 2020a).

To obtain a formally proper cost criterion, the stochastic variables $z^G(k), z^R(k)$ and virtual variables $x^G(k), x^R(k)$ are used, and, similarly to (Typaldos et al., 2020a), this yields the objective function as follows

$$J = E \left\{ \sum_{k=0}^{k_{max}-1} \left[\frac{1}{2} a(k)^2 + \left(x^G(k) + (1 - z^R(k)) x^R(k) \right) \cdot J_{DG}^*[\mathbf{x}(k), a(k), k+1] \right] \right\} \quad (6.10)$$

The recursive Stochastic Bellman Equation (SBE) has four corresponding parts. Starting from k_{max}^R , we need to move backwards, calculating the function V (optimal cost-to-go) step-by-step. The SBE for the generalized problem reads as follows

$$\begin{aligned} & V[\mathbf{x}(k), x^G(k), x^R(k), k] = \\ & = \min_{a(k) \in U} \left\{ E \left\{ \frac{1}{2} a(k)^2 + (x^g(k) + (1 - z^R(k)) x^R(k)) \cdot J_{DG}^*(\mathbf{x}(k+1), k+1) \right. \right. \\ & \quad \left. \left. + V[\mathbf{x}(k+1), x^G(k+1), x^R(k+1), k+1] \right\} \right\} = \\ & = \min_{a(k) \in U} \left\{ E \left\{ \frac{1}{2} a(k)^2 + (x^g(k) + (1 - z^R(k)) x^R(k)) \cdot J_{DG}^*(\mathbf{x}(k+1), k+1) \right. \right. \\ & \quad \left. \left. + V[\mathbf{x}(k), a(k), x^G(k) z^G(k), x^R(k) x^R(k), k+1] \right\} \right\} = \\ & = \min_{a(k) \in U} \left\{ \frac{1}{2} a(k)^2 + (x^g(k) + p^R(k) x^R(k)) \cdot J_{DG}^*(\mathbf{x}(k+1), k+1) \right. \\ & \quad \left. + (p^G(k) x^G(k) + (1 - p^R(k)) x^R(k)) V[\mathbf{x}(k), a(k), 0, 1, k+1] \right\} = \\ & \quad \left. + ((1 - p^G(k)) x^G(k)) V[\mathbf{x}(k), a(k), 1, 0, k+1] \right\} \end{aligned} \quad (6.11)$$

where

$$J_{DG}^*(\mathbf{x}(k+1), k+1) = \boldsymbol{\sigma}(\mathbf{x}(k+1)) \cdot J_{DG}^*(\mathbf{x}(k+1), k+1) \quad (6.12)$$

$$\sigma(\mathbf{x}(k+1)) = \begin{cases} 1 & \text{if } x(k+1) > 0 \text{ and } v(k+1) \leq v_{max} \text{ and } k < k_{max}^G \\ \infty & \text{if } x(k+1) > 0 \text{ and } v(k+1) > v_{max} \text{ and } k < k_{max}^G \\ 1 & \text{if } \mathbf{x}(k+1) \in \mathbf{X} \text{ and } k \geq k_{max}^G \\ \infty & \text{if } \mathbf{x}(k+1) \notin \mathbf{X} \text{ and } k \geq k_{max}^G \\ 0 & \text{else} \end{cases} \quad (6.13)$$

The details of SBE for each traffic signal phase are not explained here, due to limited space.

6.4 Preliminary Results and Discussion

In this section, preliminary results of the proposed generalized GLOSA problem considering both green and red phases are reported. Two scenarios are considered, and the obtained results are compared with those derived from IDM. The two scenarios differ in the actual switching time of the uncertain green phase, i.e., in Scenario 1 the uncertain green is extended until the latest possible time, while in Scenario 2 the green switches to red earlier. The scenarios chosen have the following set up: $x_0 = 0$ m, $v_0 = 5$ m/s, $x_e = 370$ m, $v_e = 11$ m/s and $x_1 = 300$ m. Moreover, the bounds for the states and control are set to $[x_{min}, x_{max}] = [0, 300]$ m, $[v_{min}, v_{max}] = [0, 16]$ m/s and $[a_{min}, a_{max}] = [-2, 1]$ m/s². The time step T is 1 s and the switching time windows for the traffic signal are $[k_{min}^G, k_{max}^G] = [10, 30]$ s and $[k_{min}^R, k_{max}^R] = [40, 60]$ s with uniform a-priori probability distributions. The switching from green to red for Scenarios 1 and 2 occurs at $k = k_{max}^G$ and $k = 20$, respectively. The switching from red to green at the uncertain red phase is considered to be at k_{max}^R , for both scenarios.

Fig. 6.2 displays the state (position and speed) and control (acceleration) trajectories resulting from the SDP algorithm (blue lines) and the obtained trajectories using IDM (magenta lines), for both Scenarios 1 and 2. In Fig. 6.2a-6.2c (Scenario 1), both vehicles, guided by the SDP and the IDM, cross the traffic signal before the red phase. On the other hand, in Fig. 6.2d-6.2f (Scenario 2), the vehicles do not have the time to cross before the switching and they wait until the end of the uncertain red phase. In both scenarios, the SDP algorithm, is more conservative, as it incorporates all the available knowledge, including the probabilistic distribution of the switching times and its updates over time. The proposed approach performs better in terms of fuel consumption compared to IDM, as the fuel consumption derived from the ARRB fuel consumption model (Akçelik, Biggs, 1987) is 44.2 ml. and 46.3 ml. for the SDP and IDM, respectively, for Scenario 1; and 65.3 ml. and 75.4 ml. for Scenario 2. This outcome is expected, especially in Scenario 2, where, in contrast to IDM, SDP avoids the full stop at the traffic light position. The resulting final times are $t_e = 35.2$ s and 32.0 s and $t_e = 68.5$ s and 74.0 s for SDP and IDM for Scenarios 1 and 2, respectively.

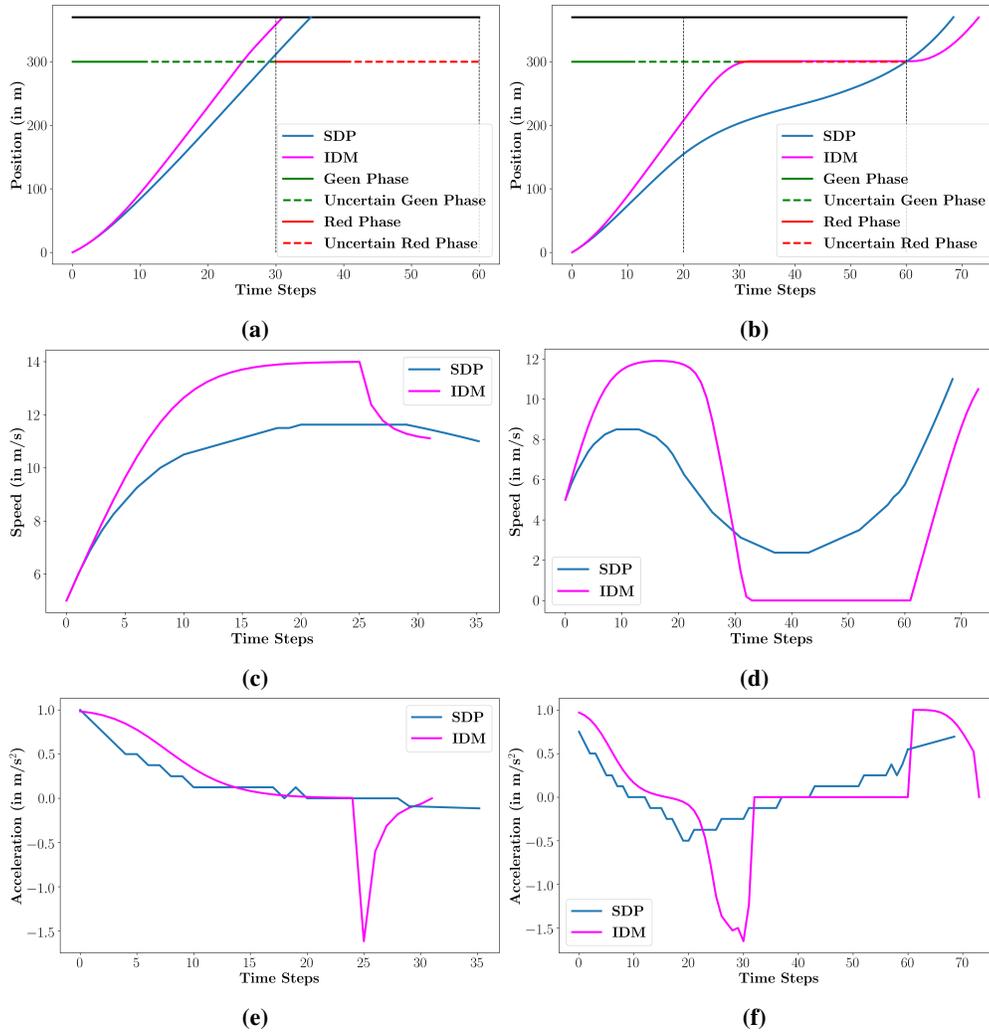


Figure 6.2: Optimal state and control trajectories of SDP (blue lines) versus IDM (magenta lines). In (a) and (d), the actual switching times are indicated with vertical dashed lines for Scenarios 1 and 2, respectively.

6.5 Conclusions

In recent work (Typaldos et al., 2020), a stochastic GLOSA methodology was developed, by optimizing, using SDP techniques, the vehicle kinematic trajectories subject to the stochastic

traffic signal switching, with fixed final state and free final time. However, the problem considered only half traffic light cycle, i.e., the traffic signal is initially red and turns to green. As an extension to that work, we consider here a full traffic light cycle, which consists of four phases, consisting of a certain green phase, an uncertain green phase, a certain red phase, and an uncertain red phase. Preliminary results illustrate the superiority of the proposed stochastic GLOSA on average, when compared with any other approach, e.g., the IDM.

Future work:

- Investigation of scenarios with multiple consecutive traffic lights and different switching times,
- Implementation of different Dynamic Programming algorithms which enable faster solutions.
- Compare the proposed SDP approach with more sophisticated approaches than IDM.

Acknowledgments

The research leading to these results has received funding from the European Research Council under the European Union's Horizon 2020 Research and Innovation programme / ERC Grant Agreement n. [833915], project TrafficFluid, see: <https://www.trafficfluid.tuc.gr>.

Chapter 7

Conclusion

In the near future, the automated vehicles' (AVs) market is expected to grow exponentially. The potential benefits of AVs are substantial, ranging from social, economic, and environmental to safety aspects. The current work contributes by delivering several methodologies and algorithms regarding the path-planning problem of AVs moving in both highway and urban environments. These contributions explore the impact of AVs on the traffic flow, while indicating their efficiency in terms of safe, comfortable and fuel-efficient driving.

Concerning the eco-driving problem, all the proposed works contribute towards the reduction of fuel consumption by minimizing the acceleration of the vehicles. Although square-of-acceleration term is commonly used for eco-driving problems, it is questionable how accurate may be, compared to more complex fuel-consumption models. To explore the effectiveness of this simple term, an optimal control problem formulation, which minimizes the fuel consumption of a vehicle, by optimizing its kinematic trajectories was proposed. For the fuel consumption estimation, a number of alternatives are employed. Initially, a realistic, but nonlinear and non-smooth formula from the literature is considered. Simple smoothing procedures are then applied, so as to enable the application of powerful numerical algorithms for the efficient solution of the resulting nonlinear optimal control problem. Furthermore, suitable quadratic approximations of the nonlinear formula are also considered, which enable analytical problem solutions. A comprehensive comparison of each alternative approach in various driving scenarios demonstrates that the square-of-acceleration term delivers excellent approximations for fuel minimizing trajectories in the present setting.

In order to investigate the effect of CAVs in freeway environments, a path-planning algorithm for connected and non-connected automated road vehicles on multilane motorways was derived from the opportune formulation of an optimal control problem. For the numerical solution of the non-convex optimization problem, a combination of NLP and DP techniques was utilized, which enables high-quality local minima. Thanks to very low computation times, the approach is readily executable within a model predictive control (MPC) framework. The proposed MPC-based approach is embedded within the Aimsun micro-simulation platform,

enabling us to thoroughly examine the behaviour of the approach in presence of vehicles emulating human driving behaviour and in a plethora of realistic driving instances. Demonstration results indicate higher efficiency of the optimally controlled vehicles in driving closer to their desired speed, compared to non-automated vehicles. In addition, increasing penetration rates of controlled vehicles are found to lead to more efficient traffic flow. Specifically, as the penetration rate of controlled vehicles rises, there is a significant increase of the average speed and, consequently, a decrease on the average delay time and travel time for all vehicles. Also, connected controlled vehicles are more efficient than non-connected controlled vehicles, due to the improved real-time information that enables more pertinent obstacle movement prediction.

To address a common dilemma raised whenever a vehicle approaches a traffic signal, i.e., whether to maintain speed or accelerate/decelerate, a deterministic and a stochastic GLOSA methodologies, by optimizing vehicles' kinematic trajectories subject to the intermediate traffic signal constraints and with a fixed final state, were proposed. For the first case, the problem was formulated as an optimal control problem and was solved analytically via PMP (Pontryagin's Maximum Principle). Subsequently, the case of stochastic switching time with known probability distribution was also addressed, and the problem was cast in the format of a stochastic optimal control problem, which was solved numerically using SDP (Stochastic Dynamic Programming). The proposed SDP algorithm may take several minutes to execute, which implies that the solution is not real-time feasible and can therefore not be obtained on-board the vehicle, but must be executed offline, at the infrastructure side, and be communicated to approaching vehicles according to their current state. As an extension of the SDP problem, a Dynamic Programming (DDDP) algorithm was developed, which solves the original SDP problem iteratively, each time considering a reduced state space. Demonstration results demonstrate that the DDDP algorithm delivers the same optimal solutions as the full range SDP, but features significantly reduced computation time. Subsequently, as an attempt to further reduce the computation times, an iterative Differential Dynamic Programming (DDP) algorithm was employed. Within each iteration, DDP solves, for each time-step separately, a quadratic-linear approximation of the recursive Bellman equation. Demonstration results indicate that the DDP algorithm strongly outperforms, both the full-range SDP and the DDDP algorithms in terms of computation time; and even delivers a slightly better optimal solution due to lack of discretisation. The implicit assumption in all pre-mentioned works, regarding the GLOSA problem, was that the traffic signal is initially red and turns to green, representing only half of the traffic signal cycle. Thus, an extension of GLOSA problem was developed, which considers a full traffic light cycle, i.e. a certain green phase; an uncertain green phase; a certain red phase; and an uncertain red phase.

In conclusion, this thesis have proposed several contributions regarding the path-planning of CAVs in both highway and urban environments. Specifically, the proposed approaches

could be beneficial to crucial aspects of road transportation, such as fuel-efficiency, safety and traffic performance. Furthermore, this work could be a good base for current and future developments and applications as, some the addressed topics are currently in an early stage, i.e the development of speed advisory systems for adaptive traffic signal; while the methodologies can be, readily, extended to be applied in potential forthcoming innovative concepts, e.g., path-planning problems for lane-free environments (Papageorgiou et al., [2021](#); Yanumula et al., [2021](#)).

Bibliography

- Ahn, K., Rakha, H., Trani, A., Van Aerde, M. (2002). “Estimating vehicle fuel consumption and emissions based on instantaneous speed and acceleration levels”. *Journal of transportation engineering* 128.2, pp. 182–190.
- Aimsun Next, . (2019). *Transport Simulation System (TSS)*. <https://www.aimsun.com/aimsun-next>.
- Akçelik, R., Biggs, D. (1987). “Acceleration profile models for vehicles in road traffic”. *Transportation Science* 21.1, pp. 36–54.
- Bar-Yehuda, R., Feldman, J. A., Pinter, R. Y., Wimer, S. (1989). “Depth-first-search and dynamic programming algorithms for efficient CMOS cell generation”. *IEEE transactions on computer-aided design of integrated circuits and systems* 8.7, pp. 737–743.
- Barkenbus, J. N. (2010). “Eco-driving: An overlooked climate change initiative”. *Energy Policy* 38.2, pp. 762–769.
- Bellman, R. (1966). “Dynamic programming”. *Science* 153.3731, pp. 34–37.
- Bernardo, M. di, Falcone, P., Salvi, A., Santini, S. (2015). “Design, analysis, and experimental validation of a distributed protocol for platooning in the presence of time-varying heterogeneous delays”. *IEEE Transactions on Control Systems Technology* 24.2, pp. 413–427.
- Bertsekas, D. (2012). *Dynamic programming and optimal control: Volume I*. Vol. 1. Athena scientific.
- Bertsekas, D. P. (1995). *Dynamic programming and optimal control*. Vol. 1. 2. Athena scientific Belmont, MA.
- Biggs, D., Akçelik, R. (1986). “Energy-related model of instantaneous fuel consumption”. *Traffic Engineering and Control* 27.6, pp. 320–325.
- Blokpoel, R., Lu, M. (2017). “Signal plan stabilization to enable eco-driving”. *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 133–138.
- Bowyer, D. P., Akçelik, R., Biggs, D. (1985). *Guide to fuel consumption analyses for urban traffic management*. 32.
- Budhiraja, R., Carpentier, J., Mastalli, C., Mansard, N. (2018). “Differential dynamic programming for multi-phase rigid contact dynamics”. *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, pp. 1–9.

- Carvalho, A., Gao, Y., Gray, A., Tseng, H. E., Borrelli, F. (2013). "Predictive control of an autonomous ground vehicle using an iterative linearization approach". *16th International IEEE conference on intelligent transportation systems (ITSC 2013)*. IEEE, pp. 2335–2340.
- Cho, G., Shaw, D. X. (1997). "A depth-first dynamic programming algorithm for the tree knapsack problem". *INFORMS Journal on Computing* 9.4, pp. 431–438.
- Chu, K., Lee, M., Sunwoo, M. (2012). "Local path planning for off-road autonomous driving with avoidance of static obstacles". *IEEE transactions on intelligent transportation systems* 13.4, pp. 1599–1616.
- Claussmann, L., Carvalho, A., Schildbach, G. (2015). "A path planner for autonomous driving on highways using a human mimicry approach with binary decision diagrams". *2015 European Control Conference (ECC)*. IEEE, pp. 2976–2982.
- Claussmann, L., Revilloud, M., Gruyer, D., Glaser, S. (2019). "A review of motion planning for highway autonomous driving". *IEEE Transactions on Intelligent Transportation Systems* 21.5, pp. 1826–1848.
- Cleveland, W. S., Grosse, E., Shyu, W. M. (2017). "Local regression models". *Statistical models in S*. Routledge, pp. 309–376.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., et al. (2001). *Introduction to algorithms, chapter 11*.
- Dey, K. C., Yan, L., Wang, X., Wang, Y., Shen, H., Chowdhury, M., Yu, L., Qiu, C., Soundararaj, V. (2015). "A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (CACC)". *IEEE Transactions on Intelligent Transportation Systems* 17.2, pp. 491–509.
- Diakaki, C., Papageorgiou, M., Papamichail, I., Nikolos, I. (2015). "Overview and analysis of vehicle automation and communication systems from a motorway traffic management perspective". *Transportation Research Part A: Policy and Practice* 75, pp. 147–165.
- Dixit, S., Fallah, S., Montanaro, U., Dianati, M., Stevens, A., Mccullough, F., Mouzakitis, A. (2018). "Trajectory planning and tracking for autonomous overtaking: State-of-the-art and future prospects". *Annual Reviews in Control* 45, pp. 76–86.
- Dixit, S., Montanaro, U., Dianati, M., Oxtoby, D., Mizutani, T., Mouzakitis, A., Fallah, S. (2019). "Trajectory planning for autonomous high-speed overtaking in structured environments using robust MPC". *IEEE Transactions on Intelligent Transportation Systems* 21.6, pp. 2310–2323.
- Earl, M. G., D'Andrea, R. (2007). "A decomposition approach to multi-vehicle cooperative control". *Robotics and Autonomous Systems* 55.4, pp. 276–291.
- Faris, W. F., Rakha, H. A., Kafafy, R. I., Idres, M., Elmoselhy, S. (2011). "Vehicle fuel consumption and emission modelling: an in-depth literature review". *International Journal of Vehicle Systems Modelling and Testing* 6.3-4, pp. 318–395.

- Fletcher, R. (1971). "A general quadratic programming algorithm". *IMA Journal of Applied Mathematics* 7.1, pp. 76–91.
- Ghiasi, A., Ma, J., Zhou, F., Li, X. (2017). "Speed harmonization algorithm using connected autonomous vehicles". *96th Annual Meeting of the Transportation Research Board*. 17–02568.
- Gilbert, E. G. (1976). "Vehicle cruise: Improved fuel economy by periodic control". *Automatica* 12.2, pp. 159–166.
- Gipps, P. G. (1986). "A model for the structure of lane-changing decisions". *Transportation Research Part B: Methodological* 20.5, pp. 403–414.
- Glaser, S., Vanholme, B., Mammari, S., Gruyer, D., Nouveliere, L. (2010). "Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction". *IEEE Transactions on intelligent transportation systems* 11.3, pp. 589–606.
- Goniewicz, K., Goniewicz, M., Pawłowski, W., Fiedor, P. (2016). "Road accident rates: strategies and programmes for improving road traffic safety". *European journal of trauma and emergency surgery* 42.4, pp. 433–438.
- González, D., Pérez, J., Milanés, V., Nashashibi, F. (2015). "A review of motion planning techniques for automated vehicles". *IEEE Transactions on Intelligent Transportation Systems* 17.4, pp. 1135–1145.
- Gu, D., Hu, H. (2002). "Neural predictive control for a car-like mobile robot". *Robotics and Autonomous Systems* 39.2, pp. 73–86.
- Gu, T., Dolan, J. M. (2014). "Toward human-like motion planning in urban environments". *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, pp. 350–355.
- Haydari, A., Yilmaz, Y. (2020). "Deep reinforcement learning for intelligent transportation systems: A survey". *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–22. DOI: [10.1109/TITS.2020.3008612](https://doi.org/10.1109/TITS.2020.3008612).
- Heidari, M., Chow, V. T., Kokotović, P. V., Meredith, D. D. (1971). "Discrete differential dynamic programming approach to water resources systems optimization". *Water Resources Research* 7.2, pp. 273–282.
- Hong, S., Malikopoulos, A. A., Lee, J., Park, B. B. (2017). "Development and evaluation of speed harmonization using optimal control theory: a simulation-based case study at a speed reduction zone 2". *Proceedings of the 96th Annual Meeting of the Transportation Research Board (TRB), Washington, DC, USA, paper*. 17-05828.
- Hooker, J. (1988). "Optimal driving for single-vehicle fuel economy". *Transportation Research Part A: General* 22.3, pp. 183–201.
- Hounsell, N., McDonald, M. (2001). "Urban network traffic control". *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* 215.4, pp. 325–334.

- Howard, T. M., Green, C. J., Kelly, A. (2010). "Receding horizon model-predictive control for mobile robot navigation of intricate paths". *Field and Service Robotics*. Springer, pp. 69–78.
- Jacobson, D. H., Mayne, D. Q. (1970). *Differential dynamic programming*. 24. Elsevier Publishing Company.
- Jalalmaab, M., Fidan, B., Jeon, S., Falcone, P. (2015). "Model predictive path planning with time-varying safety constraints for highway autonomous driving". *2015 International Conference on Advanced Robotics (ICAR)*. IEEE, pp. 213–217.
- Jamshidnejad, A., Hellendoorn, H., Lin, S., De Schutter, B. (2015). "Smoothing for efficient solution of model predictive control for urban traffic networks considering endpoint penalties". *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*. IEEE, pp. 2837–2842.
- Jamshidnejad, A., Papamichail, I., Papageorgiou, M., De Schutter, B. (2017). "Sustainable model-predictive control in urban traffic networks: Efficient solution based on general smoothing methods". *IEEE Transactions on Control Systems Technology* 26.3, pp. 813–827.
- Kalogianni, I. (2018). "Fuel-minimizing vehicle trajectory specification in the presence of traffic lights with certain or stochastic switching times". Diploma Work. School of Production Engineering and Management, Technical University of Crete, Chania, Greece.
- Kamal, M. A. S., Mukai, M., Murata, J., Kawabe, T. (2011). "Ecological vehicle control on roads with up-down slopes". *IEEE Transactions on Intelligent Transportation Systems* 12.3, pp. 783–794.
- Karafyllis, I., Theodosis, D., Papageorgiou, M. (2021). "Nonlinear adaptive cruise control of vehicular platoons". *International Journal of Control*, pp. 1–23.
- Katrakazas, C., Quddus, M., Chen, W.-H., Deka, L. (2015). "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions". *Transportation Research Part C: Emerging Technologies* 60, pp. 416–442.
- Katsaros, K., Kernchen, R., Dianati, M., Rieck, D. (2011). "Performance study of a Green Light Optimized Speed Advisory (GLOSA) application using an integrated cooperative ITS simulation platform". *2011 7th International Wireless Communications and Mobile Computing Conference*. IEEE, pp. 918–923.
- Khazaeni, Y., Cassandras, C. G. (2016). "Event-driven cooperative receding horizon control for multi-agent systems in uncertain environments". *IEEE Transactions on Control of Network Systems* 5.1, pp. 409–422.
- Kim, N., Cha, S., Peng, H. (2010). "Optimal control of hybrid electric vehicles based on Pontryagin's minimum principle". *IEEE Transactions on control systems technology* 19.5, pp. 1279–1287.

- Kitazawa, S., Kaneko, T. (2016). “Control target algorithm for direction control of autonomous vehicles in consideration of mutual accordance in mixed traffic conditions”. *Advanced Vehicle Control AVEC'16*. CRC Press, pp. 151–156.
- Kosmatopoulos, E., Papageorgiou, M., Bielefeldt, C., Dinopoulou, V., Morris, R., Mueck, J., Richards, A., Weichenmeier, F. (2006). “International comparative field evaluation of a traffic-responsive signal control strategy in three cities”. *Transportation Research Part A: Policy and Practice* 40.5, pp. 399–413.
- Koukoumidis, E., Peh, L.-S., Martonosi, M. R. (2011). “Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory”. *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pp. 127–140.
- Kuwata, Y., Fiore, G. A., Teo, J., Frazzoli, E., How, J. P. (2008). “Motion planning for urban driving using RRT”. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 1681–1686.
- Lawitzky, A., Wollherr, D., Buss, M. (2013). “Energy optimal control to approach traffic lights”. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 4382–4387.
- Loke, S. W. (2019). “Cooperative automated vehicles: A review of opportunities and challenges in socially intelligent vehicles beyond networking”. *IEEE Transactions on Intelligent Vehicles* 4.4, pp. 509–518.
- Lu, M., Ferragut, J., Kutilla, M., Chen, T. (2020). “Next-generation wireless networks for V2X”. *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 1–5.
- Ma, J., Li, X., Shladover, S., Rakha, H. A., Lu, X.-Y., Jagannathan, R., Dailey, D. J. (2016). “Freeway speed harmonization”. *IEEE Transactions on Intelligent Vehicles* 1.1, pp. 78–89.
- Ma, J., Zhou, F., Huang, Z., Melson, C. L., James, R., Zhang, X. (2018). “Hardware-in-the-loop testing of connected and automated vehicle applications: a use case for queue-aware signalized intersection approach and departure”. *Transportation Research Record* 2672.22, pp. 36–46.
- Ma, L., Xue, J., Kawabata, K., Zhu, J., Ma, C., Zheng, N. (2014). “A fast RRT algorithm for motion planning of autonomous road vehicles”. *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 1033–1038.
- Mahler, G., Vahidi, A. (2012). “Reducing idling at red lights based on probabilistic prediction of traffic signal timings”. *2012 American Control Conference (ACC)*. IEEE, pp. 6557–6562.
- Makantasis, K., Papageorgiou, M. (2018). “Motorway path planning for automated road vehicles based on optimal control methods”. *Transportation Research Record* 2672.19, pp. 112–123.

- Malikopoulos, A. A., Cassandras, C. G., Zhang, Y. J. (2018a). “A decentralized energy-optimal control framework for connected automated vehicles at signal-free intersections”. *Automatica* 93, pp. 244–256.
- Malikopoulos, A. A., Hong, S., Park, B. B., Lee, J., Ryu, S. (2018b). “Optimal control for speed harmonization of automated vehicles”. *IEEE Transactions on Intelligent Transportation Systems* 20.7, pp. 2405–2417.
- Marti, E., De Miguel, M. A., Garcia, F., Perez, J. (2019). “A review of sensor technologies for perception in automated driving”. *IEEE Intelligent Transportation Systems Magazine* 11.4, pp. 94–108.
- Mayne, D. Q. (2014). “Model predictive control: Recent developments and future promise”. *Automatica* 50.12, pp. 2967–2986.
- Mayne, D. Q., Michalska, H. (1988). “Receding horizon control of nonlinear systems”. *Proceedings of the 27th IEEE Conference on Decision and Control*. IEEE, pp. 464–465.
- Mensing, F., Trigui, R., Bideaux, E. (2011). “Vehicle trajectory optimization for application in ECO-driving”. *Vehicle Power and Propulsion Conference (VPPC), 2011 IEEE*. IEEE, pp. 1–6.
- Milanés, V., Godoy, J., Villagrà, J., Pérez, J. (2010). “Automated on-ramp merging system for congested traffic situations”. *IEEE Transactions on Intelligent Transportation Systems* 12.2, pp. 500–508.
- Monastyrsky, V., Golownykh, I. (1993). “Rapid computation of optimal control for vehicles”. *Transportation Research Part B: Methodological* 27.3, pp. 219–227.
- Montanaro, U., Dixit, S., Fallah, S., Dianati, M., Stevens, A., Oxtoby, D., Mouzakitis, A. (2019). “Towards connected autonomous driving: review of use-cases”. *Vehicle system dynamics* 57.6, pp. 779–814.
- Murillo, M., Sánchez, G., Genzelis, L., Giovanini, L. (2018). “A real-time path-planning algorithm based on receding horizon techniques”. *Journal of Intelligent & Robotic Systems* 91.3, pp. 445–457.
- Murray, D. M., Yakowitz, S. J. (1979). “Constrained differential dynamic programming and its application to multireservoir control”. *Water Resources Research* 15.5, pp. 1017–1027.
- (1984). “Differential dynamic programming and Newton’s method for discrete optimal control problems”. *Journal of Optimization Theory and Applications* 43.3, pp. 395–414.
- Nagy, B., Kelly, A. (2001). “Trajectory generation for car-like robots using cubic curvature polynomials”. *Field and Service Robots* 11.
- Nilsson, J., Falcone, P., Ali, M., Sjöberg, J. (2015). “Receding horizon maneuver generation for automated highway driving”. *Control Engineering Practice* 41, pp. 124–133.
- Ntousakis, I. A., Nikolos, I. K., Papageorgiou, M. (2016). “Optimal vehicle trajectory planning in the context of cooperative merging on highways”. *Transportation research part C: emerging technologies* 71, pp. 464–488.

- Ozatay, E., Ozguner, U., Michelini, J., Filev, D. (2014). “Analytical solution to the minimum energy consumption based velocity profile optimization problem with variable road grade”. *IFAC Proceedings Volumes* 47.3, pp. 7541–7546.
- Ozatay, E., Ozguner, U., Onori, S., Rizzoni, G. (2012). “Analytical solution to the minimum fuel consumption optimization problem with the existence of a traffic light”. *ASME 2012 5th Annual Dynamic Systems and Control Conference joint with the JSME 2012 11th Motion and Vibration Conference*. American Society of Mechanical Engineers, pp. 837–846.
- Papageorgiou, M., Diakaki, C., Dinopoulou, V., Kotsialos, A., Wang, Y. (2003). “Review of road traffic control strategies”. *Proceedings of the IEEE* 91.12, pp. 2043–2067.
- Papageorgiou, M., Leibold, M., Buss, M. (2015). *Optimierung*. Vol. 4. Springer.
- Papageorgiou, M., Marinaki, M., Typaldos, P., Makantasis, K. (2016). “A feasible direction algorithm for the numerical solution of optimal control problems – extended version”. *Technical University of Crete, Dynamics Systems and Simulations Laboratory, Chania, Greece*.
- Papageorgiou, M., Mountakis, K.-S., Karafyllis, I., Papamichail, I., Wang, Y. (2021). “Lane-free artificial-fluid concept for vehicular traffic”. *Proceedings of the IEEE* 109.2, pp. 114–121.
- Petit, N., Sciarretta, A. (2011). “Optimal drive of electric vehicles using an inversion-based trajectory generation approach”. *IFAC Proceedings Volumes* 44.1, pp. 14519–14526.
- Pontryagin, L. S. (1987). *Mathematical theory of optimal processes*. CRC Press.
- (2018). *Mathematical theory of optimal processes*. Routledge.
- Post, K., Kent, J., Tomlin, J., Carruthers, N. (1984). “Fuel consumption and emission modelling by power demand and a comparison with other models”. *Transportation Research Part A: General* 18.3, pp. 191–213.
- Qin, S. J., Badgwell, T. A. (2003). “A survey of industrial model predictive control technology”. *Control engineering practice* 11.7, pp. 733–764.
- Rahman, M., Chowdhury, M., Xie, Y., He, Y. (2013). “Review of microscopic lane-changing models and future research opportunities”. *IEEE transactions on intelligent transportation systems* 14.4, pp. 1942–1956.
- Rajamani, R. (2011). *Vehicle dynamics and control*. Springer Science & Business Media.
- Rasekhipour, Y., Khajepour, A., Chen, S.-K., Litkouhi, B. (2016). “A potential field-based model predictive path-planning controller for autonomous road vehicles”. *IEEE Transactions on Intelligent Transportation Systems* 18.5, pp. 1255–1267.
- Richter, A. (2005). “Geschwindigkeitsvorgabe an lichtsignalanlagen”. *Deutscher Universitätsverlag*.

- Rios-Torres, J., Malikopoulos, A. A. (2016a). “A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps”. *IEEE Transactions on Intelligent Transportation Systems* 18.5, pp. 1066–1077.
- (2016b). “Automated and cooperative vehicle merging at highway on-ramps”. *IEEE Transactions on Intelligent Transportation Systems* 18.4, pp. 780–789.
- Saboohi, Y., Farzaneh, H. (2009). “Model for developing an eco-driving strategy of a passenger vehicle based on the least fuel consumption”. *Applied Energy* 86.10, pp. 1925–1932.
- Sadat, A., Casas, S., Ren, M., Wu, X., Dhawan, P., Urtasun, R. (2020). “Perceive, predict, and plan: Safe motion planning through interpretable semantic representations”. *European Conference on Computer Vision*. Springer, pp. 414–430.
- Sanchez, M., Cano, J.-C., Kim, D. (2006). “Predicting traffic lights to improve urban traffic fuel consumption”. *2006 6th International Conference on ITS Telecommunications*. IEEE, pp. 331–336.
- Saxena, D. M., Bae, S., Nakhaei, A., Fujimura, K., Likhachev, M. (2020). “Driving in dense traffic with model-free reinforcement learning”. *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 5385–5392.
- Schrank, D., Lomax, T., Eisele, B., et al. (2011). “2011 urban mobility report”.
- Schwarzkopf, A., Leipnik, R. (1977). “Control of highway vehicles for minimum fuel consumption over varying terrain”. *Transportation Research* 11.4, pp. 279–286.
- Shao, Y. S., Chen, C., Kousik, S., Vasudevan, R. (2021). “Reachability-based trajectory safeguard (rts): A safe and fast reinforcement learning safety layer for continuous control”. *IEEE Robotics and Automation Letters* 6.2, pp. 3663–3670.
- Shim, T., Adireddy, G., Yuan, H. (2012). “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* 226.6, pp. 767–778.
- Singh, S. (2015). *Critical reasons for crashes investigated in the national motor vehicle crash causation survey*. Tech. rep.
- Sjoberg, K., Andres, P., Buburuzan, T., Brakemeier, A. (2017). “Cooperative intelligent transport systems in Europe: Current deployment status and outlook”. *IEEE Vehicular Technology Magazine* 12.2, pp. 89–97.
- Spiliopoulou, A., Manolis, D., Vandorou, F., Papageorgiou, M. (2018). “Adaptive cruise control operation for improved motorway traffic flow”. *Transportation research record* 2672.22, pp. 24–35.
- Stahlmann, R., Möller, M., Brauer, A., German, R., Eckhoff, D. (2016). “Technical evaluation of glosa systems and results from the field”. *2016 IEEE Vehicular Networking Conference (VNC)*. IEEE, pp. 1–8.

- Sun, C., Guanetti, J., Borrelli, F., Moura, S. J. (2020). “Optimal eco-driving control of connected and autonomous vehicles through signalized intersections”. *IEEE Internet of Things Journal* 7.5, pp. 3759–3773.
- Tassa, Y., Mansard, N., Todorov, E. (2014). “Control-limited differential dynamic programming”. *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1168–1175.
- Teng, H., Yu, L., Qi, Y. (2002). “Statistical microscale emission models incorporating acceleration and deceleration”. *81st transportation research board annual meeting*. Vol. 29.
- Tian, D., Wu, G., Boriboonsomsin, K., Barth, M. J. (2018). “Performance measurement evaluation framework and co-benefit/tradeoff analysis for connected and automated vehicles (CAV) applications: A survey”. *IEEE Intelligent Transportation Systems Magazine* 10.3, pp. 110–122.
- Treiber, M., Kesting, A. (2013). “Traffic flow dynamics”. *Traffic Flow Dynamics: Data, Models and Simulation*, Springer-Verlag Berlin Heidelberg, pp. 983–1000.
- Troullinos, D., Chalkiadakis, G., Papamichail, I., Papageorgiou, M. (2021). “Collaborative multiagent decision making for lane-free autonomous driving”. *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 1335–1343.
- Turri, V., Besselink, B., Johansson, K. H. (2016). “Cooperative look-ahead control for fuel-efficient and safe heavy-duty vehicle platooning”. *IEEE Transactions on Control Systems Technology* 25.1, pp. 12–28.
- Typaldos, P. (2017). “Minimization of fuel consumption for vehicle trajectories”. MA thesis. School of Production Engineering & Management.
- Typaldos, P., Kalogianni, I., Mountakis, K. S., Papamichail, I., Papageorgiou, M. (2020a). “Vehicle trajectory specification in presence of traffic lights with known or uncertain switching times”. *Transportation research record* 2674.8, pp. 53–66.
- Typaldos, P., Koutsas, P., Papamichail, I., Papageorgiou, M. (2022a). “GLOSA system with uncertain green and red signal phases”. *4th Symposium on Management of Future motorway and urban Traffic Systems (MFTS 2022)*.
- Typaldos, P., Papageorgiou, M. (2022). “Modified dynamic programming algorithms for GLOSA systems with stochastic signal switching times”. URL: <https://arxiv.org/abs/2211.12159>.
- Typaldos, P., Papageorgiou, M., Papamichail, I. (2022b). “Optimization-based path-planning for connected and non-connected automated vehicles”. *Transportation Research Part C: Emerging Technologies* 134, p. 103487.
- Typaldos, P., Papamichail, I., Papageorgiou, M. (2020b). “Minimization of fuel consumption for vehicle trajectories”. *IEEE Transactions on Intelligent Transportation Systems* 21.4, pp. 1716–1727.

- Typaldos, P., Volakakis, V., Papageorgiou, M., Papamichail, I. (2021). “Vehicle-based trajectory specification in presence of traffic lights with stochastic switching times”. *IFAC-PapersOnLine* 54.2, pp. 298–305.
- Van Arem, B., Van Driel, C. J., Visser, R. (2006). “The impact of cooperative adaptive cruise control on traffic-flow characteristics”. *IEEE Transactions on intelligent transportation systems* 7.4, pp. 429–436.
- Van De Hoef, S., Johansson, K. H., Dimarogonas, D. V. (2017). “Fuel-efficient en route formation of truck platoons”. *IEEE Transactions on Intelligent Transportation Systems* 19.1, pp. 102–112.
- Van Leersum, J. (1985). “Implementation of an advisory speed algorithm in transyt”. *Transportation Research Part A: General* 19.3, pp. 207–217.
- Van Mierlo, J., Maggetto, G., Van de Burgwal, E., Gense, R. (2004). “Driving style and traffic measures-influence on vehicle emissions and fuel consumption”. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 218.1, pp. 43–50.
- Wang, H., Huang, Y., Khajepour, A., Liu, T., Qin, Y., Zhang, Y. (2018a). “Local path planning for autonomous vehicles: Crash mitigation”. *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, pp. 1602–1606.
- Wang, Y., Li, X., Yao, H. (2018b). “Review of trajectory optimisation for connected automated vehicles”. *IET Intelligent Transport Systems* 13.4, pp. 580–586.
- Wang, Z., Bian, Y., Shladover, S. E., Wu, G., Li, S. E., Barth, M. J. (2019). “A survey on cooperative longitudinal motion control of multiple connected and automated vehicles”. *IEEE Intelligent Transportation Systems Magazine* 12.1, pp. 4–24.
- Wang, Z., Wu, G., Barth, M. J. (2017a). “Developing a distributed consensus-based cooperative adaptive cruise control system for heterogeneous vehicles with predecessor following topology”. *Journal of Advanced Transportation* 2017.
- (2018c). “A review on cooperative adaptive cruise control (CACC) systems: Architectures, controls, and applications”. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 2884–2891.
- Wang, Z., Wu, G., Hao, P., Boriboonsomsin, K., Barth, M. (2017b). “Developing a platoon-wide eco-cooperative adaptive cruise control (CACC) system”. *2017 IEEE Intelligent Vehicles symposium (IV)*. IEEE, pp. 1256–1261.
- Werling, M., Ziegler, J., Kammel, S., Thrun, S. (2010). “Optimal trajectory generation for dynamic street scenarios in a frenet frame”. *2010 IEEE International Conference on Robotics and Automation*. IEEE, pp. 987–993.
- Wolfram Research, Inc. (2016). *Mathematica, Version 11.0*.

- Xie, Z., Liu, C. K., Hauser, K. (2017). “Differential dynamic programming with nonlinear constraints”. *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 695–702.
- Xu, S., Deng, K., Li, S. E., Li, S., Cheng, B. (2014). “Legendre pseudospectral computation of optimal speed profiles for vehicle eco-driving system”. *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, pp. 1103–1108.
- Yakowitz, S. (1986). “The stagewise Kuhn-Tucker condition and differential dynamic programming”. *IEEE transactions on automatic control* 31.1, pp. 25–30.
- Yang, H., Almutairi, F., Rakha, H. (2020). “Eco-driving at signalized intersections: A multiple signal optimization approach”. *IEEE Transactions on Intelligent Transportation Systems* 22.5, pp. 2943–2955.
- Yang, H., Rakha, H., Ala, M. V. (2016). “Eco-cooperative adaptive cruise control at signalized intersections considering queue effects”. *IEEE Transactions on Intelligent Transportation Systems* 18.6, pp. 1575–1585.
- Yanumula, V. K., Typaldos, P., Troullinos, D., Malekzadeh, M., Papamichail, I., Papageorgiou, M. (2021). “Optimal path planning for connected and automated vehicles in lane-free traffic”. *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, pp. 3545–3552.
- Zhao, W., Ngoduy, D., Shepherd, S., Liu, R., Papageorgiou, M. (2018). “A platoon based cooperative eco-driving model for mixed automated and human-driven vehicles at a signalised intersection”. *Transportation Research Part C: Emerging Technologies* 95, pp. 802–821.
- Zhao, Z., Wang, Z., Wu, G., Ye, F., Barth, M. J. (2019). “The state-of-the-art of coordinated ramp control with mixed traffic conditions”. *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, pp. 1741–1748.