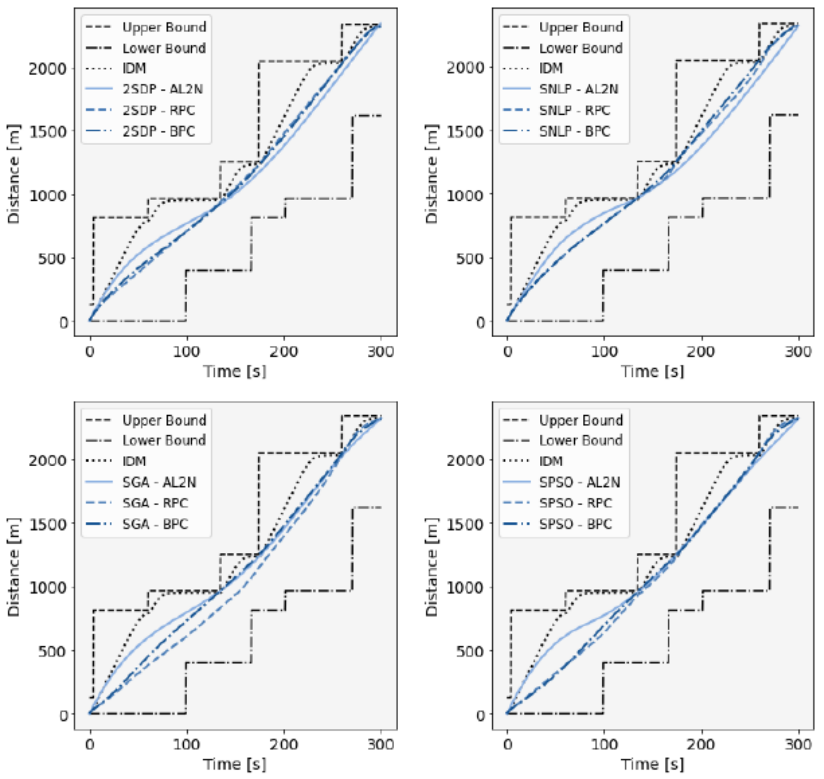
MOUNTAIN-PLAINS CONSORTIUM

MPC 24-558 | A. Rabinowitz, C. Ang, Y. Mahmoud, F. Araghi, R. Meyer
I. Kolmanovsky, Z. Asher, and T.H. Bradley

REAL-TIME IMPLEMENTATION COMPARISON OF URBAN ECO-DRIVING CONTROLS



A University Transportation Center sponsored by the U.S. Department of Transportation serving the Mountain-Plains Region. Consortium members:

Colorado State University
North Dakota State University
South Dakota State University
University of Colorado Denver
University of Denver
University of Utah
Utah State University
University of Wyoming

Technical Report Documentation Page

1. Report No.
MPC-570

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
Real-Time Implementation Comparison of Urban Eco-Driving Controls

5. Report Date
September 2024

6. Performing Organization Code

7. Author(s)
Aaron I. Rabinowitz, Chon Chia Ang, Yara Hazem Mahmoud,
Farhang Motallebi Araghi, Richard T. Meyer, Ilya Kolmanovsky,
Zachary D. Asher, Thomas H. Bradley

8. Performing Organization Report No.
MPC 24-558

9. Performing Organization Name and Address
Colorado State University
Department of Systems Engineering
6029 Campus Delivery
Fort Collins, CO 80523-6029

10. Work Unit No. (TRAIS)

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
Mountain-Plains Consortium
North Dakota State University
PO Box 6050, Fargo, ND 58108

13. Type of Report and Period Covered
Final Report

14. Sponsoring Agency Code

15. Supplementary Notes
Supported by a grant from the US DOT, University Transportation Centers Program

16. Abstract
Connected autonomous vehicle (CAV) technology has the potential to enable significant gains in energy economy (EE). Much research attention has been focused on autonomous eco-driving control enabled by various methods. In this study, the state of the literature on autonomous eco-driving control is reviewed, an overall system's description of eco-driving control for a CAV is provided, and representative methods are evaluated comparatively against each other in simulation. Simulations are conducted using real-world traffic signal data and a validated future automotive systems technology simulator (FASTSim) model. Results indicate that an EE improvement in the range of 5%–15% is attainable depending on the method and cost function used. This article shows that dynamic programming (DP) methods are most effective in improving EE but are significantly more computationally expensive than other methods. The genetic algorithm (GA) methods are shown to present the most potential in terms of EE improvement and run-time. Results also indicate that velocity-sensitive cost functions allow all the methods to perform better than pure acceleration minimization.

17. Key Word
advanced traffic management systems, connected vehicles, fuel consumption, infrastructure, intelligent vehicles, lagrangian functions, sensors, sustainable transportation, trajectory, vehicle safety, velocity

18. Distribution Statement
Public distribution

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
37

22. Price
n/a

Real-Time Implementation Comparison of Urban Eco-Driving Controls

Aaron I. Rabinowitz, Graduate Research Assistant
Colorado State University

Chon Chia Ang, Graduate Research Assistant
Colorado State University

Yara Hazem Mahmoud, Graduate Research Assistant
Western Michigan University
Kalamazoo MI

Farhang Motallebi Araghi, Graduate Research Assistant
Western Michigan University
Kalamazoo MI

Richard T. Meyer, Associate Professor
Western Michigan University
Kalamazoo MI

Ilya Kolmanovsky, Professor
University of Michigan
Ann Arbor, MI

Zachary D. Asher, Associate Professor
Western Michigan University
Kalamazoo MI

Thomas H. Bradley, Professor
Colorado State University
Department of Systems Engineering
6029 Campus Delivery
Fort Collins, CO 80523-6029

September 2024

Acknowledgements

This work was supported in part by the U.S. Department of Energy under Grant DE-EE0008468; and in part by the Colorado State University and the Mountain-Plains Consortium, a University Transportation Center funded by the U.S. Department of Transportation.

Disclaimer

“The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.”

North Dakota State University does not discriminate in its programs and activities on the basis of age, color, gender expression/identity, genetic information, marital status, national origin, participation in lawful off-campus activity, physical or mental disability, pregnancy, public assistance status, race, religion, sex, sexual orientation, spousal relationship to current employee, or veteran status, as applicable. Direct inquiries to Vice Provost, Title IX/ADA Coordinator, Old Main 100, (701) 231-7708, ndsu.eoaa@ndsu.edu.

ABSTRACT

Connected autonomous vehicle (CAV) technology has the potential to enable significant gains in energy economy (EE). Much research attention has been focused on autonomous eco-driving control enabled by various methods. In this study, the state of the literature on autonomous eco-driving control is reviewed, an overall system's description of eco-driving control for a CAV is provided, and representative methods are evaluated comparatively against each other in simulation. Simulations are conducted using real-world traffic signal data and a validated future automotive systems technology simulator (FASTSim) model. Results indicate that an EE improvement in the range of 5%–15% is attainable depending on the method and cost function used. This article shows that dynamic programming (DP) methods are most effective in improving EE but are significantly more computationally expensive than other methods. The genetic algorithm (GA) methods are shown to present the most potential in terms of EE improvement and run-time. Results also indicate that velocity-sensitive cost functions allow all the methods to perform better than pure acceleration minimization.

TABLE OF CONTENTS

1. INTRODUCTION 1

2. LITERATURE REVIEW 2

2.1 Rule-Based Eco-Driving 2

2.2 Discretized Control Optimization 2

2.3 Polynomial Trajectory Optimization 3

2.4 Summary 5

3. SYSTEM DEFINITION 6

4. SUBSYSTEM 1: PERCEPTION 7

4.1 Path Constraints 7

5. SUBSYSTEM 2: PLANNING 9

5.1 Baseline Control 9

5.2 Optimal Control 10

6. SUBSYSTEM 3: PLANT 15

7. RESULTS 16

7.1 Optimal Solver Results 16

7.2 Eco-Driving Traces 19

7.3 Summary 21

8. CONCLUSION 23

9. REFERENCES 24

LIST OF TABLES

Table 2.1 Publications Reviewed by Method Type 5

Table 4.1 Data Available to CAVS 7

Table 5.1 Variables And Parameters For IDM 9

Table 5.2 EE Regression Results For IDM Parameters..... 10

Table 5.3 Kia Soul EV FASTSim Model Parameters..... 14

LIST OF FIGURES

Figure 3.1 Eco-driving system schematic 6

Figure 4.1 Example of a “corridor” with upper and lower boundaries 8

Figure 5.1 Comparison of cost function values for 2015 Kia Soul EV. Red polygon outlines the operational envelope of the UDDS, US06, and HWFET EPA dynamometer drive cycles and is shown as reference for common driving conditions 12

Figure 7.1 Mean and standard deviation of EE improvement over baseline results for all the methods and cost functions 16

Figure 7.2 Mean and standard deviation of cost function reduction over baseline results for all the methods and cost functions 17

Figure 7.3 Correlation between cost function reduction and EE improvement for all the cost functions 17

Figure 7.4 Significance of comparative results (P-values), purple indicates that the column significantly outperformed the row, blue indicates that the row significantly outperformed the column, and green indicates the difference between the row and column 18

Figure 7.5 Mean and standard deviations of run-times for all methods and cost functions 18

Figure 7.6 Example position versus time traces for all the methods and cost functions 20

Figure 7.7 Example velocity versus time traces for all the methods and cost functions 21

Figure 7.8 Comparison of run-time and EE improvement means for all the methods and cost functions 21

EXECUTIVE SUMMARY[1]

This report investigates the efforts to enhance vehicle energy efficiency (EE) and reduce greenhouse gas (GHG) emissions from road vehicles, as driven by climate change concerns and energy costs. It examines the transition from traditional internal combustion engines to hybrid and battery electric vehicles (BEVs) and how these technologies promise further GHG reductions. The study focuses on eco-driving strategies, particularly the effectiveness of autonomous eco-driving, which can be implemented more easily and at scale compared with manual strategies, thereby circumventing driver training issues.

A key part of the report is a comprehensive comparative study that addresses the lack of such analysis in existing literature. This study summarizes eco-driving control strategies, sets up a framework for comparing solver methods, and evaluates these methods using real-world data. The literature review categorizes methods into rule-based and optimal, with optimal further divided into globally optimal methods and locally optimal methods. It discusses the advantages of autonomous eco-driving over manual strategies, including precision, scalability, and driver acceptance.

The report’s methodology involves a detailed review of the literature, system definition, and analysis of eco-driving subsystems, including perception, planning, and the plant subsystem. It explains the computational models used for simulations and compares various control methods such as rule-based eco-driving, discretized control optimization, and polynomial trajectory optimization.

The results section details the effectiveness of each method in terms of EE improvement and computational load, highlighting that while globally optimal solutions (like those from dynamic programming) offer the highest potential for EE improvement, they are computationally expensive. Genetic algorithms (GAs) emerge as a promising real-time method that strikes a balance between EE improvement and computational feasibility.

The report concludes that autonomous eco-driving control can significantly enhance vehicle EE with a potential national impact on energy savings if widely adopted. It recommends the use of a GA method with a road power cost function as the best trade-off for generating optimal eco-driving traces for urban BEVs.

This paper was previously published as Rabinowitz, et al., “Real-time implementation comparison of Urban EcoDriving Controls.” In *IEEE Transactions on Control Systems Technology*, Vol 32, no. 1 pp 143-157, 2024

[1] Executive summary generated using OpenAI. (2024). *ChatGPT* (Feb 7 version) [Large language model]. https://chat.openai.com/chat

1. INTRODUCTION

In response to rising concerns over climate change and energy costs, a significant portion of automotive development effort has gone into the reduction in energy use and greenhouse gas (GHG) emissions from road vehicles. Over time, vehicles have become significantly more efficient in terms of both energy economy (EE) and GHG emissions/mi [1], [2] under pressure from environmental regulations from the U.S. Environmental Protection Agency (EPA) and its global equivalents, which exert ongoing pressure on original equipment manufacturers (OEMs) to continue this effort [3]. To improve vehicular energy efficiency, traditional internal combustion engine (ICE) powertrains have incorporated electric motors and evolved into hybrid electric vehicles and battery electric vehicles (BEVs), [4] which promise further GHG reductions per vehicle [5]. Regardless of powertrain technology and regardless of methods of power generation, the pressure to reduce vehicular energy consumption will continue to be present.

Vehicle energy efficiency is also subject to modes of operation. Eco-driving is a strategy designed to reduce fuel consumption by minimizing accelerations and unnecessary braking events. Eco-driving is well known and has been shown to be effective when used by human drivers [6]. As an example, eco-driving is taught as a part of drivers' education in Singapore and has resulted in an EE improvement of 11% to 15% there [7]. Differences in culture, infrastructure, and available technology will play a major role in determining the effectiveness of efforts to promote manual eco-driving. Vehicular autonomy and connected autonomous vehicle (CAV) technology provide a more general opportunity for the application of eco-driving strategies because they circumvent driver acceptance/training issues. When compared with a human driver (i.e., manual eco-driving), a CAV has the ability to follow optimal trajectories precisely and can consider information that is beyond line-of-sight.

Compared with manual eco-driving, autonomous eco-driving yields the following potential benefits:

1) ability to precisely follow optimal energy traces
2) ability to account for traffic information that is beyond line-of-sight
3) ease and scalability of implementation
4) improved driver/passenger acceptance

A great variety of solutions for autonomous eco-driving control have been put forward in the literature. This diversity is due to the complicated nature of the problem and the many dimensional design spaces that results from it. To the author's knowledge, no comprehensive, comparative study exists. This study attempts to address this research gap by summarizing and subdividing eco-driving control strategies, defining a framework for comparative implementation of solver methods, implementing a selection of common methods, and evaluating these methods in terms of performance and practicality using real-world data [8]. The current state of the literature is discussed in Section 2. A system and subsystem overview for an assumed eco-driving CAV is provided in Sections 3–6. The results are presented in Section 7 and conclusions in Section 8.

2. LITERATURE REVIEW

Much research exists in the area of autonomous eco-driving controls. In conducting the literature review, the authors were particularly interested in publications that proposed methods which might be implemented in real time. A real-time control was defined as one that was explicitly or could be implemented in a receding-horizon setting. Such a control should be able to execute multiple times per second.

The authors propose that the methods reviewed may be categorized by purpose and structure as follows. First, a division can be made into the categories of rule-based and optimal methods. The rule-based methods provide simple and computationally light algorithms for computing target speed on an instantaneous basis. The rule-based methods often mimic the heuristics that human drivers follow when attempting to minimize energy consumption such as lighter accelerations and longer following distances. In contrast, optimal methods attempt to find a minimum energy consumption trace for a given time or distance horizon. Optimal methods thus require information about future conditions even if this is done purely with assumptions. Within the set of optimal methods, one can further subdivide into globally optimal and locally optimal methods. The globally optimal methods find the control that results in the global minimum energy consumption. For the globally optimal methods, function dictates form, and all the methods proposed are variants of dynamic programming (DP). The locally optimal methods find a control trace more efficient than one that could be attained by a rule-based method but require less computational time than the globally optimal methods. The locally optimal methods often involve transcribing the problem into the time domain and performing trajectory optimization. As will be seen in Section 7, local optima will often resemble the global optimum far more closely than a rule-based method solution. The authors propose a taxonomy based on groupings in form and function which divides methods into the following categories: rule-based eco-driving (RBED), discretized control optimization (DCO), and polynomial trajectory optimization (PTO).

2.1 Rule-Based Eco-Driving

RBED is a subset of autonomous eco-driving control wherein a vehicle reduces its energy consumption through a set of predefined rules, which are functions of vehicle states. Because of their feedforward nature, the RBED methods are relatively simple to implement. Compared with normal human driving behavior, the RBED methods are capable of yielding considerable fuel economy improvement [9], [10]. A common RBED algorithm is an intelligent driver model (IDM) [11] with several works presenting modified versions of the method in eco-driving simulations [12], [13], [14], [15]. Although non-IDM RBED methods appear in [16], [17], and [18], IDM and its derivatives dominate the RBED literature and are often used as a comparison point in optimal eco-driving literature. When implemented on a sufficient percentage of vehicles, the RBED methods have shown promise in traffic calming [12], [19]. RBED control has also been extended to cooperative and centralized fleet control schemes [13], [14].

2.2 Discretized Control Optimization

The purpose of a DCO method is to compute optimal controls for a vehicle at a set of discrete points in time or distance. The DCO methods require a state transition model and information about future exogenous inputs. The DCO category primarily consists of the DP and reinforcement learning (RL) methods.

The DP [20], [21] is a well-known mathematical optimization method that produces globally optimal solutions to control problems subject to a chosen discretization. A realization of the DP-derived optimal solution depends on whether the chosen discretization and the model appropriately match the real-world application. To account for constraints in position and speed inherent to autonomous eco-driving control, both must be problem states. The control in the autonomous eco-driving problem is acceleration or a related control such as throttling. Such a two-state one-control DP algorithm, presented in [22] and [23], minimizes fuel consumption while navigating around traffic signals. Hellström et al. [24] present a two-state one-control DP algorithm for heavy-duty trucks in highway conditions. Both methods must execute at a low rate and serve to set targets for a lower level controller. The primary issue with the DP methods for real-time implementation is that run-times scale exponentially with the number of states and controls. This scaling issue is often referred to as the "curse of dimensionality." In the autonomous eco-driving literature, DP solutions proposed as real-time controls use suboptimal implementations of DP to avoid the issue. The DP methods are also often proposed as a high-level control algorithm, executing at low frequency, which serves to set targets for a low-level controller. It is most common to see DP implemented as a comparison point for the performance of another proposed solution with the caveat that the DP solution is not a candidate for real-time implementation.

Suboptimal implementations of DP are found in [25], [26], [27], and [28]. Maamria et al. [25] and [26] overcome the run-time scaling issues by removing position as a problem state. This is accomplished by adding a tunable constant cost to the running cost to ensure that the correct final distance is reached at the correct time. This tunable parameter must be found via numerical root-finding. Overall, this method can be thought of as a pseudo two-state DP (2SDP) method. The pseudo two-state method was found to execute in less time than an equivalent 2SDP method, which emphasizes the importance of the run-time scaling effects inherent to DP. A major limitation with [25] and [26] is that having removed the position state, it is not possible for the optimization to account for traffic signals in fixed positions, making the method less applicable for urban eco-driving. Deshpande et al. [27] propose an approximate DP (ADP) solver for traffic-signal constrained driving, which uses a non-optimal rollout method to approximate the cost-to-go. Deshpande et al. [27] account for traffic signals by determining whether it is feasible to pass in a "go" phase or, if not, implementing eco-approach and eco-departure. Gupta et al. [28] propose a method by which precomputed DP solutions may be adjusted to account for perturbations in external inputs without having to recompute the DP solution, thus reducing the required frequency of DP method evaluations for real-time control. In all the cases, global optimality is traded for reductions in run-time.

Vahidi and Sciarretta [29], Stanger and del Re [30], Xu and Peng [31], Groelke et al. [32], Bae et al. [33], and Sun et al. [34] propose a DP-based method where the DP solution is computed at a low frequency and is used as a target by a lower level controller. Exemplary of the type is [35], which uses vehicle to infrastructure (V2I) information and DP to set velocity targets for a cruise control system for urban driving. This method was tested both in hardware in loop (HIL) simulation and on-road and was shown to produce a 30% EE improvement at a cost of an 8% increase in travel time.

The RL-based methods are proposed in [36], [37], and [38]. Lee et al. [36] use RL for optimizing motor power control for an electric vehicle subject to road grade but not traffic. The RL control was found to perform nearly as well as DP for the same problem. The algorithms seen in [37] and [38] are focused on comfort (reduction in jerk) and collision avoidance rather than eco-driving, and also found similar performance to equivalent DP solutions with lower run-time. Ultimately, RL suffers from the same disadvantages that DP does for the application: the long run-time required to compute the strategy but not to the same extent.

2.3 Polynomial Trajectory Optimization

The optimal eco-driving optimal control problem can also be solved as a trajectory optimization problem by transcribing into the time domain. Direct transcription transforms the problem into an *n*-dimensional optimization with the number of dimensions set by the level of discretization but at lower levels of discretization. Run-time for the trajectory optimization will scale with dimensionality depending on the solver used. At very high levels of discretization, linear interpolation can be used between trajectory points. To reduce run-time, a coarser discretization may be used but this will necessitate polynomial interpolation between the optimization points. Because every segment of an interpolation polynomial is a function of multiple knot points, using an interpolation polynomial comes at the cost of introducing nonlinearity into the problem. The PTO methods may use bounded nonlinear solvers such as interior-point optimization (IPOPT), sequential least-squares programming (SLSQP), or metaheuristics.

PTO is commonly used for motion planning in robotics [39]. Nonlinear bounded solvers are used to perform PTO for autonomous eco-driving in [40], [41], and [42]. A comparison to DP is provided in [43] for the related optimal energy management problem where the PTO method, using a nonlinear bounded solver, was able to approximate the globally optimal solution and to produce a solution in orders of magnitude in less time than DP. The constraints of the optimal eco-driving problem are difficult for bounded nonlinear solvers to deal with. The issue is that vehicle motion is subject to time-varying constraints in position caused by other vehicles and traffic signals, as well as the speed of other vehicles and speed limits. These constraints will be discussed in Section 4. The combination of nonlinearity caused by interpolation polynomials and the complexity of the constraints makes the computation of meaningful gradients difficult, and thus, gradient-based solvers could struggle. Hamednia et al. [40], Khalik et al. [41], and Padilla et al. [42] do not consider distance and speed constraints simultaneously. The issues that gradient-based solvers experience are somewhat mitigated in [44], in which best interpolation splines [45], [46] are used rather than interpolation polynomials. Best interpolation splines consider each segment separately and can be used to guarantee that constraints will not be violated, but this is at the cost of additional run-time.

Metaheuristics are also commonly used as solvers for PTO methods with genetic algorithm (GA) and particle swarm optimization (PSO) methods being the most often proposed. GA and PSO take inspiration from nature. PSO [47] takes inspiration from animals that exhibit schooling or swarming behaviors. PSO works by generating a field of candidate solutions (particles) and then computing gradients for each particle based on individual and global best discovered solutions. GA [48], [49], [50] mimics natural selection by encoding decision variables for a discretized problem into phenotypes and then mating the highest fitness phenotypes over many successive iterations. GA can be modified in many ways, such as introducing random mutation, elitist selection, and others to change the breadth of the search. Where PSO is still at its core a gradient descent method, GA is not and therefore not subject to the difficulties that gradient descent methods face with the optimal eco-driving problem.

GA was used in [51] to generate optimal driving operations from real-world data with the final results yielding an improved fuel economy of 22% compared with the initial population. Similarly, Li et al. [52] used GA to group vehicles in compatible streams to provide a smoother traffic flow with the algorithm scaling favorably in comparison to DP. In recent studies [53], [54], [55], GA PTO methods were applied to both conventional and electric vehicles with results showing favorable fuel economy improvement for both types of vehicles. PSO was used in various studies to optimize energy consumption for individual vehicles [52], [56], [57], [58] and to streamline vehicle platoon behavior at intersections [59]. A comparison of PSO-based PTO methods with DP [58] found that PSO significantly underperformed DP in terms of efficiency but executed in significantly less time. PSO and GA have also been used in combination, and the combined method was shown to be more effective than individually [52], [57].

Table 2.1 Publications Reviewed by Method Type

Category	Method Type	Publications
Rules-Based Eco-Driving (RBED)	Intelligent Driver Model (IDM)	[9], [10], [11], [12], [19], [13], [14], [15]
	Rule-Set	[16], [17], [18]
Discretized Control Optimization (DCO)	Dynamic Programming (DP)	[22], [23], [25], [26], [27], [28] [29], [30], [31], [32], [33], [34], [35]
	Reinforcement Learning (RL)	[36], [37], [38]
Polynomial Trajectory Optimization (PTO)	Bounded Nonlinear Solvers	[40], [41], [42], [44]
	Metaheuristics	[51], [52], [53], [54], [55] [52], [56], [57], [58]

The general consensus in the literature would be that PTO methods provide the opportunity to compute locally optimal solutions in substantially less time than a globally optimal solution could be computed using DP. The constraints used in much of the PTO literature were simplifications of what the authors would consider the minimum constraints for optimal eco-driving in urban conditions. The complex boundaries inherent to the optimal eco-driving problem are difficult for gradient-based solvers to account for and are, perhaps, easier for metaheuristics to account for. However, the use of GA or PSO comes at the cost of introducing randomness to the problem.

2.4 Summary

The publications reviewed are listed by category and method type in Table 2.1. The literature contains a variety of approaches to the optimal eco-driving problem. There is significant variation in the constraints used in the studies surveyed. The distinction in constraints more or less reflects a division in focus between urban driving and other types of driving. Urban driving is constrained by the positions and velocities of surrounding vehicles as well as traffic signal locations and states. The constraints present in urban driving are time-varying. Inevitably, the constraints used will need to be approximate as precise knowledge of future values is not possible. Because all the optimal eco-driving methods proposed are intended to be used in a receding-horizon manner, some simplification is acceptable. However, to make direct comparisons between methods, a standard and sufficiently representative set of constraints must be applied to all.

3. SYSTEM DEFINITION

The eco-driving system can be broken down into three subsystems, as shown in Figure 3.1. This system-level diagram is consistent with advanced vehicle control applications such as autonomous vehicles [60] and with energy efficiency improvement strategies such as optimal energy management [61]. The eco-driving subsystems are, respectively, the perception subsystem, the planning subsystem, and the plant subsystem.

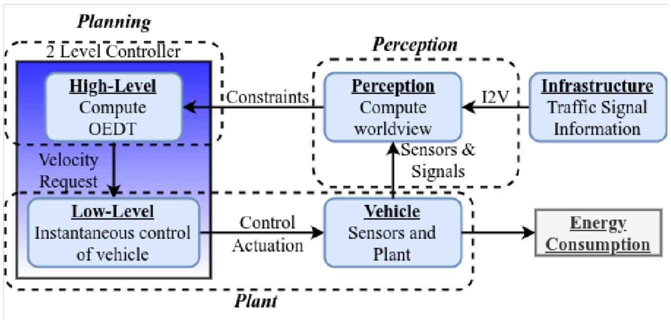


Figure 3.1 Eco-driving system schematic

The perception subsystem uses the sensors and connectivity capabilities of the ego vehicle and computes motion boundaries based on a detected lead vehicle (with on-board sensors and V2V) and upcoming traffic signal information (V2I). The proliferation of new vehicles equipped with a forward object detection system will soon reach 100% per an agreement between the National Highway Traffic Safety Administration (NHTSA) and automakers, which mandates the inclusion of said systems to enable automatic emergency braking as a standard feature [62]. These systems often comprise a radar and a visual object detection system, which work in concert to determine the location, motion, and type of objects in the ego vehicle's forward vision cone. In addition to enabling safety-oriented features such as collision avoidance systems, the forward object detection system also enables convenience-oriented features such as adaptive cruise control. In the future, most vehicles may also be equipped with V2I technology in the form of a transponder, which communicates with infrastructure transponders according to the Society of Automotive Engineers (SAE) specification J2735 [63]. Among the messages contained in the SAE J2735 specifications are the SPaT and MAP messages, which provide the signal phase and timing and locations of the upcoming traffic signals.

The vehicle is assumed to contain a two-level controller with a high-level controller computing an optimal eco-driving trace, and the low-level controller being responsible for carrying out the optimal eco-driving trace in a safe manner. The planning subsystem, which is composed of the high-level controller, takes the information about lead vehicle motion, speed limit, and future traffic signal information and uses it to compute the optimal eco-driving trace.

The final subsystem, the plant, is the ego vehicle (physical or simulated), which executes the optimal eco-driving trace and outputs the resultant energy consumption. The subsystems and the manner in which they are treated in this study are explained in Sections 4–6.

4. SUBSYSTEM 1: PERCEPTION

For the eco-driving control to be evaluated in a real-world context, the algorithms that generate the optimal eco-driving trace must be able to function using only information that is currently or will soon be available to CAVs. The information available to CAVs comes from the advanced driver assistance system (ADAS) system of the CAV and from V2I communication where available. The data, which are available to CAVs via their ADAS systems and V2I communication, are listed in Table 4.1 and further elaborated in [64].

Table 4.1 Data Available to CAVS

Signal	Description	Source
Lead Vehicle	Relative location of confirmed lead vehicle	ADAS
Signal Phase and Timing (SPaT)	Phase and timing for subsequent traffic signals	V2I
Positions of Subsequent Traffic Lights (MAP)	Latitude and longitude coordinates for subsequent traffic signals	V2I
Speed Limit	Currently active speed limit for the ego vehicle	V2I

With this information, a CAV can generate path constraints for the optimal eco-driving problem. For this study, path constraints consist of allowable locations (distances along the vehicle path) for the vehicle at specific times and allowable speeds for the vehicle at specific locations.

4.1 Path Constraints

The ego vehicle should not be deliberately programmed to violate traffic laws even if this provides efficiency and/or travel-time benefits [65]. This means that the vehicle should not exceed the speed limit, disobey traffic signals, or collide with any other vehicle. If the ego vehicle is the first vehicle in a queue, then an upper boundary can be generated from SPaT knowledge as an inequality constraint

$$x(t) < B_U(t), \quad t \in [0, T] \qquad (1)$$

where x is the vehicle distance along its route, B_U is the upper boundary which is a function of time, and T is the final time of the drive cycle. There are many ways to generate this upper boundary based on lead vehicle and traffic signal state. A general approach would be to formulate the upper boundary based on a piecewise function wherein the boundary is generated by the closer of the nearest stop phase and the immediate lead vehicle. For the purposes of this study, only vehicles with no immediate lead vehicle are considered (i.e., there are vehicles in front of the ego vehicle but always at least one signal away) as in such a case, a long-term optimal trajectory can be generated.

An assumption made here is that waiting out a go phase while not moving, although potentially optimal for a single vehicle, will cause congestion and will not be fleet optimal. Thus, a lower bound on distance as a function of time is also defined as an inequality constraint

$$x(t) > B_L(t), \quad t \in [0, T] \qquad (2)$$

where B_L is a piecewise function of time based on the positions and phases of leading traffic signals. The upper and lower bounds combine to form a “corridor” on a phase map, as shown in Figure 4.1.

In limiting possible paths to those entirely within the corridor, the ego vehicle is limited to largely following traffic norms and is far less likely to radically affect normal traffic patterns. The selection of stop phases to define the boundaries is done using the IDM model, which is further described in Section 5.

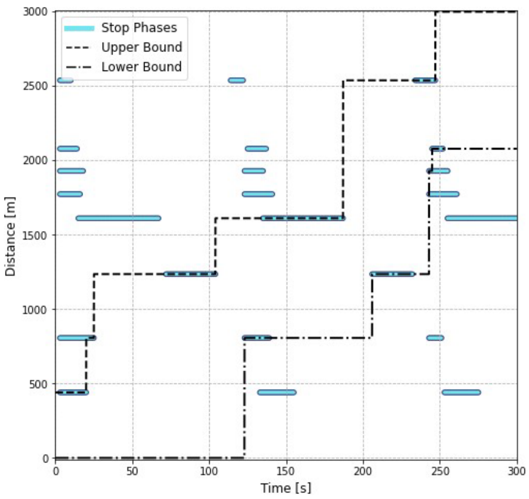


Figure 4.1 Example of a “corridor” with upper and lower boundaries

The IDM simulation is carried out for a given amount of time, and the upper bound is defined by the preceding phases of the traffic signals passed by the model vehicle, and the lower bound is defined by succeeding phases of the same signals. As the IDM model represents a baseline driver, the corridor created this way is one that must reflect normal driving and, thus, is useful for this purpose. Note that the boundaries created in this manner are nonconvex.

Traffic signals are generally adaptive as they were in this specific case. In this study, full knowledge of traffic signal timing in the future is assumed. The effects of adaptive traffic signal timing may be dealt with through the implementation of stochastic constraints as in [34]. Uncertainty about the timing of traffic signals will have the effect of extending the stop phases as used in optimization and thus tightening the corridor.

An element of reality is added to this study through the use of real-world SPaT data in the generation of path boundaries. These data were collected in 2019 and consist of traffic light phase and timing data from 19 traffic signals along a four-mile route in downtown Fort Collins, CO. These data were collected by the authors and their collection is described in [66]. Several hours of SPaT data for each of the traffic signals were collected in collaboration with the Fort Collins Traffic Operations Center. From these data and the distances of the traffic signals along the route, a phase map was constructed.

To conform to traffic norms and regulations, the ego vehicle velocity is required to satisfy the inequality

$$0 \leq v(t) \leq S_L(t), \quad t \in [0, T] \qquad (3)$$

where $S_L(t)$ is the road speed limit at time t. For the Fort Collins drive cycle used in this study, the speed limit for all roads at all times was 35 mi/h (15.65 m/s).

5. SUBSYSTEM 2: PLANNING

The planning subsystem is responsible for calculating an optimal eco-driving trace based on the constraints computed by the perception subsystem. As described in Section 3, the planning system is assumed to contain a high-level controller, which computes optimal velocities, and a low-level controller, which implements them. This study is only concerned with the high-level controller. It is also assumed that the high-level controller will operate with the shrinking horizon and the solution recomputed at discrete time instants. The optimal methods selected for implementation were DP, DCO, and PTO using IPOPT, GA, and PSO as solvers. IDM was used as the baseline control to compare against. These methods are defined in the following subsections.

Table 5.1 Variables And Parameters For IDM

Parameter	Description	Representative Value
i	Ego vehicle (lead vehicle is vehicle i-1)	N/A
x	Distance	N/A
v	Velocity	N/A
s	Distance headway (space between lead and follow vehicle)	N/A
s^*	Desired distance headway	N/A
η	Proportion of maximum acceleration used	N/A
s_0	Minimum distance headway	15 m
T	Desired time headway	4 s
δ	Velocity exponent	4
l	Vehicle length	2 m
a	Maximum forward acceleration	5 m/s^2
b	Maximum deceleration	5 m/s^2

5.1 Baseline Control

1) Intelligent Driver Model: IDM, developed by Trieber et al. [11] is an RBED method intended to enable agent-based traffic modeling. This model represented a step improvement in previous car-following models as it was meta-stable, prevented collisions, and all the parameters had physical interpretations. The IDM is formulated as follows with parameters described in Table 5.1:

$$\dot{x}_i = \frac{dx_i}{dt} = v_i \tag{4}$$

$$\eta = \left(1 - \left(\frac{v_i}{v_0}\right)^{\delta} - \left(\frac{s^*(v_i, v_{i-1})}{s_i}\right)^{2}\right) \tag{5}$$

$$\dot{v}_i = \frac{dv_i}{dt} = \begin{cases} a\eta, & \eta \geq 0 \\ b\eta, & \eta < 0 \end{cases} \tag{6}$$

$$s_i = x_{i-1} - x_i - l_{i-1} \tag{7}$$

$$s^*(v_i, v_{i-1}) = s_0 + v_i T + \frac{v_i(v_{i-1} - v_i)}{2\sqrt{ab}}. \tag{8}$$

In this study, as mentioned previously, only the optimal trace for the lead vehicle is considered. Thus, the upper bound of the traffic light constraints is used in place of a lead vehicle with varying distances but always with zero speed.

IDM can represent a spectrum of drivers in terms of aggression in acceleration and following distance. Parameter selection for IDM is important as it affects the efficiency of the generated trace. Those parameters with the greatest effect on EE are a, b, and δ. An experiment was run on said parameters using 100 different constraint sets per case and a future automotive systems technology simulator (FASTSim) [67] 2015 Kia Soul electric vehicle (EV) model.

Table 5.2 EE Regression Results For IDM Parameters

coef	value	std err	t	$P > \|\|t\|\|$
Intercept	143.6204	0.664	216.293	0.000
a	-1.6560	0.514	-3.220	0.005
b	-1.6921	0.514	-3.290	0.004
$a : b$	0.3316	0.398	0.832	0.416
δ	-1.6026	0.514	-3.116	0.006
$a : \delta$	-0.4282	0.398	-1.075	0.296
$b : \delta$	-0.0932	0.398	-0.234	0.817
$a : b : \delta$	0.0664	0.309	0.215	0.832

This experiment was a full-factorial design with the levels for a and b being 1, 5, and 9 m/s^2 (this range encompassing virtually all the passenger vehicle accelerations [68], [69]), and the levels for δ being 2, 4, and 6. The EE results of this experiment were regressed onto the values for a, b, and δ and interaction terms, and the results are presented in Table 5.2.

The results of the regression analysis indicated that a, b, and δ were significant terms, which negatively affected EE, while none of the interaction terms was significant. Thus, values for a, b, and δ can be set independently. Several papers propose methods for setting these values or the values themselves. In the literature, the default value for δ is given as 4 [11], [70], [71], [72]. NREL produced a report in 2021 [73], which extracted 39,000 individual driving features (acceleration-from-stop, deceleration-to-stop, and cruise events) from collected driving data and fitting IDM parameters to the data. Although the IDM model used by NREL is slightly differently formulated than in this article, the results are, nevertheless, informative. NREL found clusters for δ at 0.88, 1.40, 1.75, 2.13, and 4.78; ultimately, the article recommends a value of 4 for δ. Setting values for a and b was also based on literature where default values are generally given as 5 m/s^2 for both. The authors adopted these established values.

5.2 Optimal Control

All the optimal control solver methods address the following problem:

$$\min_{\overline{U}} \; J(S_0, \overline{U}) \qquad (9)$$

where

$$J(S_0, \overline{U}) = \Phi(S_N) + \sum_{k=1}^{N} \Psi(S_k, U_k) \qquad (10)$$

s.t.

$$S_{k+1} = f(S_k, U_k), \quad k = 0, \ldots, N-1 \qquad (11)$$

$$B_L(t) \leq S(t) \leq B_U(t) \qquad (12)$$

where $\psi(S,U)$ is the running cost, $\Phi(S)$ the final state cost, $S = [x,v]^\top$ is the state vector containing the problem states' position and velocity, $S_0 = [x_0, v_0]^\top$ is the initial values of the state vector, $U = [a]$ is the control vector containing the control acceleration, J is the cost for S and U, and B_L and B_U are the vectors containing the constraints as described in Section 4. The overline indicates a sequence of values at multiple discrete time intervals. The goal of the optimization* is to find the optimal eco-driving trace (U) such that the corresponding cost is equal to J^*.

The cost J is evaluated in one of three ways. It is common in the literature for the cost function for an eco-driving optimization to be entirely based on acceleration. The case for using an acceleration-based cost function is made in several papers [74], [75], [76]. Nevertheless, the authors chose to consider other, progressively less abstracted cost functions as well. The cost functions were also designed to enable fast computations for both long traces and for single time steps. The cost functions used are specified below.

1) Cost Functions:

a) Acceleration l^2 norm (Al2N) cost function: The Al2N cost function is simply the square of the l^2 norm of acceleration sequence. It is given by

$$J_{\mathrm{AL2N}}(\overline{U}) = \sum_{k=1}^{N} a_k^2. \qquad (13)$$

Note that minimizing the l^2 norm squared is equivalent (gives the same acceleration sequence) as minimizing the l^2 norm but is computationally advantageous as it does not require the computation of square roots and dealing with non-smoothness at the origin.

b) Road power cost (RPC) cost function: The RPC cost function is based on the road loads' ABC formula [77] multiplied by velocity to return power. This cost function considers the impacts of viscous and aerodynamic drag in addition to acceleration, and it is given by

$$J_{\mathrm{RPC}}(\overline{S}, \overline{U}) = \sum_{k=1}^{N} [Av_k + Bv_k^2 + Cv_k^3 + ma_k v_k] \qquad (14)$$

where A, B, and C are the coefficients of the road loads' equation, and m is the vehicle mass. For FASTSim vehicles, the road loads' coefficients are not provided, and hence were chosen as $A = 0$, $B = C_{\mathrm{RR}}$, and $C = \rho F C_D$ with C_{RR} being the coefficient of rolling resistance, ρ being the density of air, F being the vehicle frontal area, and C_D being the vehicle coefficient of aerodynamic drag. One of the important aspects of Al2N and RPC cost functions is their independence of powertrain model. An approach related to RPC has been studied in [78] under the name wheel power minimization.

c) Battery power cost (BPC) cost function: The BPC cost function is an extension of the RPC cost function, which accounts for the efficiency of the motor/inverter based on power requirements. This calculation is a simplified facsimile of the FASTSim model and requires powertrain modeling details. The BPC cost is calculated as follows:

$$J_{\text{BPC}}(\overline{S}, \overline{U}) = \sum_{k=1}^{N} \eta_k(v_k, a_k) P_R(v_k, a_k) \tag{15}$$

$$P_R(v_k, a_k) = A v_k + B v_k^2 + C v_k^3 + m a_k v_k \tag{16}$$

$$\eta_k(v_k, a_k) = \begin{cases} \eta_T^{-1} \eta_{M/I}^{-1}(P_R(v_k, a_k)), & P_R \geq 0 \\ \eta_T \eta_{M/I}(P_R(v_k, a_k)), & P_R < 0. \end{cases} \tag{17}$$

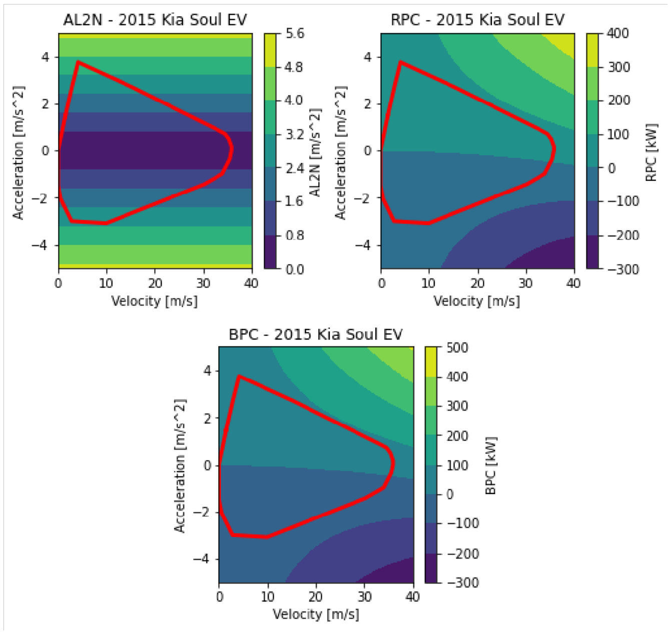


Figure 5.1 Comparison of cost function values for 2015 Kia Soul EV. Red polygon outlines the operational envelope of the UDDS, US06, and HWFET EPA dynamometer drive cycles and is shown as reference for common driving conditions

The transmission efficiency term η_T is a constant, and the motor/inverter efficiency term $\eta_{M/I}$ is calculated by interpolating using the FASTSim motor efficiency curve. Note that BPC requires more component-specific information and interpolation and thus may be more difficult to implement and requires more computational time.

d) Summary: The three different cost functions reflect three different approaches to optimizing EE. A comparison of the cost function values for a 2015 Kia Soul EV is shown in Figure 5.1. Note that the velocity-sensitive cost functions RPC and BPC have similar contour plots, but both differ significantly from the Al2N cost function.

2) Optimizers:

a) Two-state DP: DP is a well-known and commonly used optimal control method. The principal advantage of DP is that it guarantees a globally optimal solution subject to the chosen discretization. The primary disadvantage of DP is that it will generally require significantly greater computational time and effort than other methods and heuristics. In the case of eco-driving control, which is a two-state one-control nonlinear optimization problem with time-varying constraints, 2SDP is a natural choice and it appears in the literature in multiple forms, as described in Section 2. The dynamics of the problem in discrete time are represented by

$$v_{k+1} = v_k + u_k \Delta t \tag{18}$$

$$x_{k+1} = x_k + v_k \Delta t. \tag{19}$$

The boundary violation cost function J_{PC} is shown in (20), where the path constraints in x were enforced by a squared error penalty function. The boundary violation cost is added to the running cost $\psi(S_k, U_k)$ at each time step

$$J_{\mathrm{PC}}(S_k, U_k) + \begin{cases} \beta_x(x_k - B_{L,k})^2, & x_k < B_{L,k} \\ 0, & B_{L,k} \leq x_k \leq B_{U,k} \\ \beta_x(x_k - B_{U,k})^2, & x_k > B_{U,k} \end{cases}$$
$$+ \begin{cases} \beta_v(v_k - 0)^2, & v_k < 0 \\ 0, & 0 \leq v_k \leq S_{L,k} \\ \beta_v(v_k - S_{L,k})^2, & v_k > S_{L,k} \end{cases} \qquad (20)$$

where $B_{L,k} = B_L(t_k)$, $B_{U,k} = B_U(t_k)$, and $S_{L,k} = S_L(t_k)$. The final state cost function is

$$\Phi(S_N) = \beta_{\mathrm{FS}}(x_N - x_{\mathrm{target}})^2 \qquad (21)$$

where x_{target} is the desired ending position, and β_{FS} is a tuned parameter.

b) Spline nonlinear programming (SNLP): A second common method to solve time-varying control problems is via direct transcription [79], wherein a problem in continuous time is transcribed to the time domain and solved at discreet times. Direct transcription (DT) significantly increases the dimensionality of a control problem but allows the use of efficient methods such as IPOPT and SLSQP for linear and nonlinear problems [80], [81], [82]. The dynamics of the problem are represented by

$$\overline{v}_{k+1} = \overline{v}_k + \overline{u}_k \Delta t \qquad (22)$$
$$\overline{x}_{k+1} = \overline{x}_k + \overline{v}_k \Delta t. \qquad (23)$$

The running cost and final state cost functions are the same as that for 2SDP and are shown in (20) and (21), respectively.

An issue with using IPOPT to solve discrete time optimal control problems is that the run-time required scales exponentially with the length of the state vector [83]. To avoid using extremely high levels of discretization, it is common to use polynomial interpolation between more distant optimization points. The authors chose to define trajectories using piecewise cubic hermitic interpolation polynomial (PCHIP) splines with knots placed at those points in time where the upper or lower boundaries change. The trajectories are defined as

$$\overline{S} = \mathrm{PCHIP}(\overline{\epsilon}, \overline{t}_{\mathrm{knots}}, \overline{B}_{L,\mathrm{knots}}, \overline{B}_{U,\mathrm{knots}}, \overline{t}) \qquad (24)$$

where ϵ are the locations of the vehicle at the knot times (t_{knots}) relative to the boundaries at the knot times ($B_{L,\mathrm{knots}}$ and $B_{U,\mathrm{knots}}$), and t is the discrete time vector for the problem.

c) Spline GA (SGA): The first metaheuristic method discussed is the SGA. For this study, the phenotypes optimized are ϵ vectors. The initial population is generated randomly with an initial guess inserted in place of one randomly generated phenotype. The GA method uses sorted selection, wherein the best phenotypes are selected for crossover and random mutation, and wherein a certain percentage of the total chromosomes from all the phenotypes are changed to a random number at each step. The method also uses elitist carry-over, wherein the best phenotype is kept for the next step unchanged. The dynamics and cost function for SGA are identical to those for SNLP. GA is inherently parallelizable and scalable, meaning that it is well-suited to modern parallel computing and may benefit significantly in terms of run-time from such an implementation.

Table 5.3 Kia Soul EV FASTSim Model Parameters

Parameter	Value	Units	Source
Mass	1664	kg	ANL D^3 test data sheet [84]
Frontal Area	2.87	m	Public information [86]
Coefficient of Drag (C_D)	.35	N/A	Public information [86]
Coefficient of Rolling Resistance (C_{RR})	.0188	N/A	Tuned parameter
Maximum Battery Storage	25.5	kWh	Tuned Parameter
Wheelbase	2.57	m	Public information [86]
Max Motor Power	81	kW	Public information [86]

d) Spline PSO (SPSO): The second heuristic method used is SPSO, which uses the PSO heuristic to optimize a positional spline trajectory. In this study, the particles used are ϵ vectors for a given set of boundaries, and the trace in distance and velocity is computed as in (24). PSO is a quasi-Newton method as it applies a modified gradient search but does so with many particles simultaneously. The particle position and velocity update equations for PSO are given as

$$\overline{\epsilon}_{k+1} = \overline{\epsilon}_k + \overline{V}_k \tag{25}$$

$$\overline{V}_{k+1} = w\overline{V}_k + c_1 r_1(\overline{\epsilon}_{\text{best},p,k} - \overline{\epsilon}_k) + c_2 r_2(\epsilon_{\text{best},g,k} - \overline{\epsilon}_k) \tag{26}$$

where V is the vector of particle n-dimensional velocities, w is the momentum term which sets the weight of the current velocity, c_1 and c_2 are the local and global position weights, r_1 and r_2 are random weights assigned to the local and global terms, $\epsilon_{\text{best},p}$ is a vector of the best solutions found by each particle, and $\epsilon_{\text{best},g}$ is the global best solution found by any of the particles. In this study, a mutation step was added to the PSO solver to enable faster convergence [56] with the mutation step functioning similar to how it functions in the SGA method. Like GA, PSO is inherently parallelizable and scalable, making it well-suited for a parallel implementation.

6. SUBSYSTEM 3: PLANT

For this study, a 2015 Kia Soul EV was selected because dynamometer data for it are available from ANL's downloadable dynamometer database (D^3) [84] and because the research group owns a drive-by-wire capable physical vehicle for future studies. For vehicle simulation, NREL's FASTSim [67] was selected. FASTSim is an efficient, accurate, and robust longitudinal vehicle simulation commonly used in research. Construction of the FASTSim Kia Soul EV model was done using a combination of publicly available data, common FASTSim validated model parameters [85], and tuned model parameters. The model parameters are shown in Table 5.3.

The two tuned parameters, C_{RR} and maximum battery storage, were tuned from assumed values to best match the battery state of charge (SoC) and battery power traces from the D^3 data. After tuning the data, the 2015 Kia Soul EV FASTSim model matched the D^3 data to within 0.2% in terms of energy consumption while closely matching the SoC and battery power traces with mean absolute percentage error values of 0.763% and 1.552%.[2] The vehicle plant will add uncertainty to the optimization in the forms of sensor noise and actuator error. Neither of these is modeled in FASTSim. However, Motallebiaraghi et al. [87] performed a dynamometer validation study, which confirmed that similar results could be obtained with a physical vehicle plant implying that the control is robust to the noise originating from physical sensors and actuators.

[2] Zero-valued points omitted.

7. RESULTS

Each optimal eco-driving trace generation method was evaluated in terms of the following two criteria:

1) Ability to produce energy-efficient solution traces
2) Ability to produce solutions within acceptable levels of run-time

The authors evaluated the performance of the methods in generating optimal eco-driving traces for five-minute driving trajectories. Longer time horizons allow for the solvers to improve over baseline to a greater degree, but they also increase the optimization space for the solvers, leading to rapid growth in run-times. The five-minute time horizon was picked as a sufficient compromise. Although, ultimately, any on-board implementation must be receding-horizon-based to account for changing information in real-time, this report is only concerned with the efficacy of solver methods for single evaluations.

The purpose of this study was, specifically, to compare the relative merits of several optimal eco-driving trace methods found in the literature. Consequently, the scope was limited. The authors assumed that optimal eco-driving trace generation is one step in an optimal eco-driving control algorithm that operates in a receding-horizon manner, and that this algorithm comprises the upper level of a two-level controller with the lower level being responsible for the instantaneous control of the vehicle. This conception of an optimal eco-driving control framework is consistent with the literature as described in Section 2.

7.1 Optimal Solver Results

1) EE Improvement: A standard experiment was run for evaluation of the methods with respect to the solver and cost function. This experiment was a full-factorial design in which each solver was evaluated for 100 predefined boundaries cases and with each cost function. These predefined cases were defined by a selection of random starting times and locations on the phase map, as shown in Section 4. The decision to run 100 cases per combination was made to allow for the use of large sample statistics.

The results of the experiment in terms of EE improvement over baseline and in terms of cost function reduction over baseline are shown in Figure 7.1 and Figure 7.2.

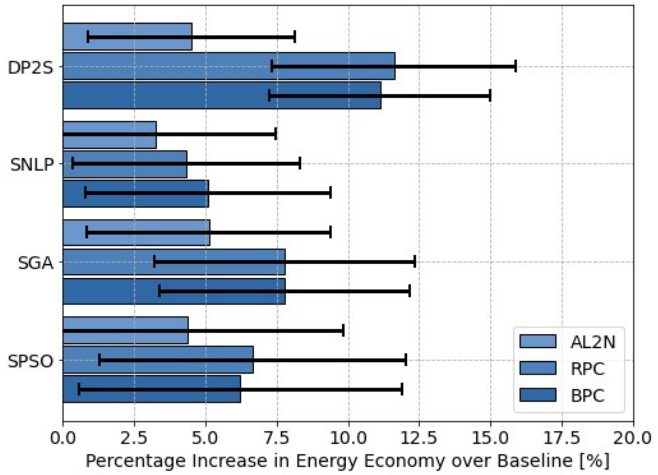


Figure 7.1 Mean and standard deviation of EE improvement over baseline results for all the methods and cost functions

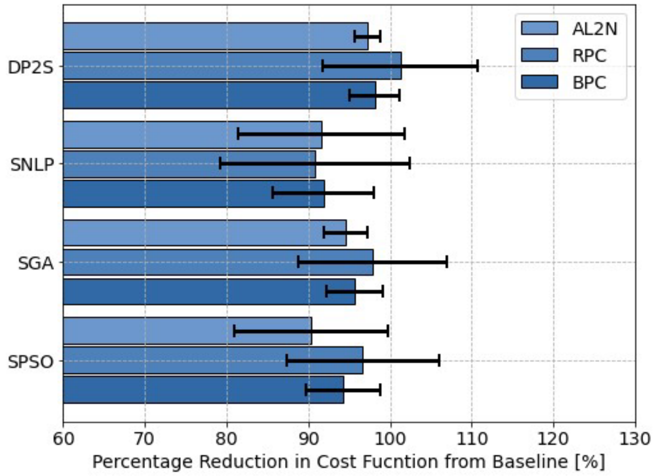


Figure 7.2 Mean and standard deviation of cost function reduction over baseline results for all the methods and cost functions

From Figure 7.1 and Figure 7.2, a definitive order is visible in the relative performances of the methods in relation to EE improvement and cost function reduction. It can be observed that the cost function reduction exceeded 100% on a recurring basis for the RPC and BPC cost functions. The ability of the RPC and BPC cost functions to be reduced by greater than 100% is reflective of the regeneration potential over a given drive cycle for those cost functions, and is an artifact of the particular boundary conditions used in the experiments. All the optimal eco-driving traces in the experiment start at 15.65 m/s (35 mi/h), which is the speed limit of the four streets used for data collection, but optimal eco-driving traces were not required to match this speed at the end of the drive cycle. Thus, it was possible for the energy regenerated over the course of the drive cycle to exceed the energy spent. Generally, the ranges seen for EE improvement as a percentage of the mean were quite large in comparison with the same for cost function improvement, and this is the result of the low correlations between cost function improvement and EE improvement for all the methods and cost functions. Correlations between cost function improvement and EE improvement are shown for all the cost functions in Figure 7.3.

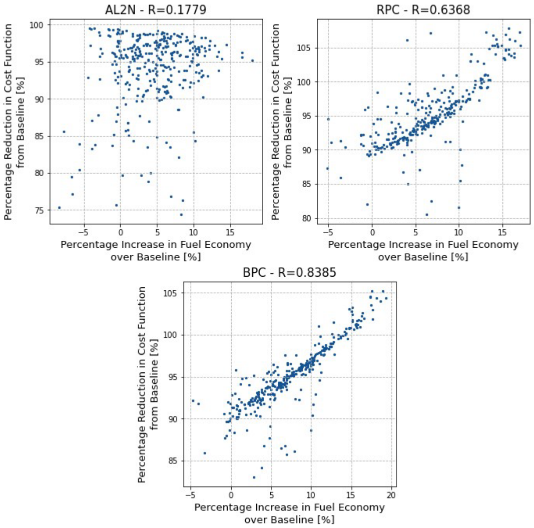
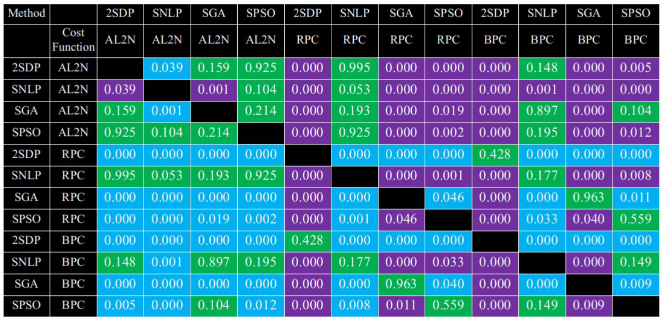


Figure 7.3 Correlation between cost function reduction and EE improvement for all the cost functions



Method		2SDP	SNLP	SGA	SPSO	2SDP	SNLP	SGA	SPSO	2SDP	SNLP	SGA	SPSO
	Cost Function	AL2N	AL2N	AL2N	AL2N	RPC	RPC	RPC	RPC	BPC	BPC	BPC	BPC
2SDP	AL2N		0.039	0.159	0.925	0.000	0.995	0.000	0.000	0.000	0.148	0.000	0.005
SNLP	AL2N	0.039		0.001	0.104	0.000	0.053	0.000	0.000	0.000	0.001	0.000	0.000
SGA	AL2N	0.159	0.001		0.214	0.000	0.193	0.000	0.019	0.000	0.897	0.000	0.104
SPSO	AL2N	0.925	0.104	0.214		0.000	0.925	0.000	0.002	0.000	0.195	0.000	0.012
2SDP	RPC	0.000	0.000	0.000	0.000		0.000	0.000	0.000	0.428	0.000	0.000	0.000
SNLP	RPC	0.995	0.053	0.193	0.925	0.000		0.000	0.001	0.000	0.177	0.000	0.008
SGA	RPC	0.000	0.000	0.000	0.000	0.000	0.000		0.046	0.000	0.000	0.963	0.011
SPSO	RPC	0.000	0.000	0.019	0.002	0.000	0.001	0.046		0.000	0.033	0.040	0.559
2SDP	BPC	0.000	0.000	0.000	0.000	0.428	0.000	0.000	0.000		0.000	0.000	0.000
SNLP	BPC	0.148	0.001	0.897	0.195	0.000	0.177	0.000	0.033	0.000		0.000	0.149
SGA	BPC	0.000	0.000	0.000	0.000	0.000	0.000	0.963	0.040	0.000	0.000		0.009
SPSO	BPC	0.005	0.000	0.104	0.012	0.000	0.008	0.011	0.559	0.000	0.149	0.009	

Figure 7.4 Significance of comparative results (P-values), purple indicates that the column significantly outperformed the row, blue indicates that the row significantly outperformed the column, and green indicates the difference between the row and column

Correlation between cost function improvement and EE improvement was shown to be best for BPC and then RPC; in both the cases, the correlation was significantly better than for Al2N. This is attributed to models in BPC and RPC providing a closer match to the model in FASTSim. Due to the large uncertainties regarding the EE improvement results, the significance of the observed differences in effectiveness could not be assumed; thus, *T*-tests were conducted between all the combinations of method and cost function, and the results are presented in Figure 7.4.

The results shown in Figures 4 and 7 indicate that the best performing method in terms of improving EE was 2SDP followed by the heuristic methods and finally SNLP. The same results indicate that the velocity sensitive cost functions enable better solutions to be found than Al2N. Neither result is surprising; only DP should be able to find globally optimal solutions, and more information should lead to a better solution.

Finally, by reducing the acceleration and braking limits by a factor of 10–0.5 m/s^2, the IDM method was able to produce a mean improvement of 5.51% with a standard deviation of 2.79%, comparable to the results from the SPSO method. The IDM method was omitted from the comparison as it cannot be made to meet the end position constraint and, thus, achieved better EE by reducing average speed and rendering the results not directly comparable.

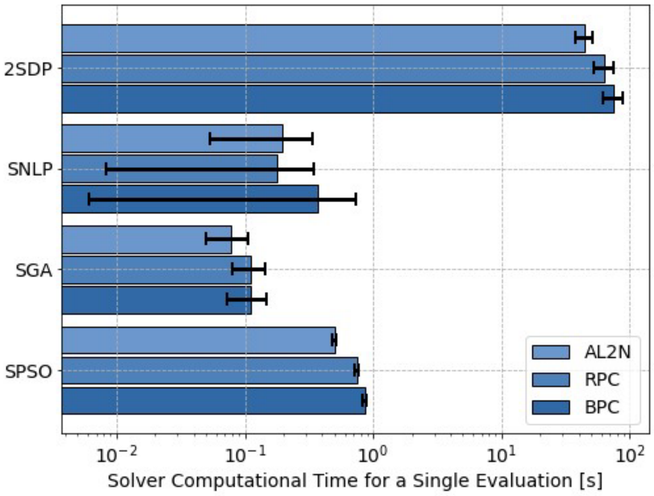


Figure 7.5 Mean and standard deviations of run-times for all methods and cost functions

2) Computational Load: All the methods for this study were implemented in Python 3 with the NumPy and Scipy libraries. All the solvers were run-time optimized in Python, and all are vectorized to the highest degree possible to minimize run-time [88]. Nevertheless, a specific outcome of the Python implementation is that Python has very limited parallel processing capability [89], which means that the authors were not able to experiment on the impacts of parallel processing on run-time for the SGA and SPSO methods. The computer used for simulation contained an AMD Ryzen 7 3700 × 8-core multithreading capable CPU with 16 GB of RAM running the 64 bit Ubuntu 18.04 LTS operating system with Python 3.8. All the simulations were conducted on the same computer to ensure the integrity of relative run-times. Even though Python is unlikely to be used for onboard implementation, the computational time results are of interest for the relative comparison of different methods in terms of computational time required. Figure 7.5 shows the relative run-times for each method and cost function.

An immediate conclusion is that the 2SDP method is not competitive with the other methods as a real-time control due to its large run-time requirement. Implementation specifics play a huge part in determining run-time, and it is possible to significantly reduce the run-time requirements for the 2SDP method by changing hardware and language; but these changes would also benefit the other methods, and the relative gap should remain within the same order of magnitude. The authors did not find a single paper in the literature which implemented a 2SDP method in real time. The closest examples would be [33] and [35], in which a DP solver is used as the higher level in a two-level receding-horizon controller, but the DP algorithm takes multiple seconds to produce a novel solution, and [27], which implements a real-time ADP solver.

It is also evident that the SGA method is the quickest to execute and could be made to execute in even less time with the use of parallel processing, with the same being true for SPSO. Current vehicular computing systems differ in architecture from desktop computers although this may soon change [90]. Implementation on automotive controllers may result in changes to the relative run-times. However, the order is unlikely to change given that the differences are in orders of magnitude.

7.2 Eco-Driving Traces

The differences in EE improvement reflect visual differences in optimal eco-driving trace traces. A representative example is shown in Figure 7.6 and Figure 7.7 for all the methods and cost functions with one set of constraints.

In general, the optimal eco-driving traces improve over the baseline traces primarily by minimizing the speed reduction due to traffic signals. There are many local optima in the results' space, and many are very similar to the global optimum; thus, the non-DP methods are most likely to settle on a local optimum. However, these local optima clearly approximate the global optimum. The optimal traces for the Al2N cost function are visually distinct from those generated using the speed-sensitive cost functions. While the Al2N cost function is only sensitive to absolute acceleration, the speed-sensitive cost functions are sensitive to directional acceleration and proportional to speed, speed squared, and speed cubed. The result is that the speed-sensitive cost functions will tend to reduce maximum speed and encourage deceleration to a greater degree than Al2N. When compared with the literature, traces seen in this study are more jerky. There are two reasons for this. First, the constraints used in this study are more complex than those used in most of the literature being time-varying and in distance and speed. The second is that no explicit proxy for passenger comfort was added to the cost function. The velocity-sensitive cost functions resulted in traces with lower speeds and higher decelerations, which would undoubtedly be less comfortable for passengers.

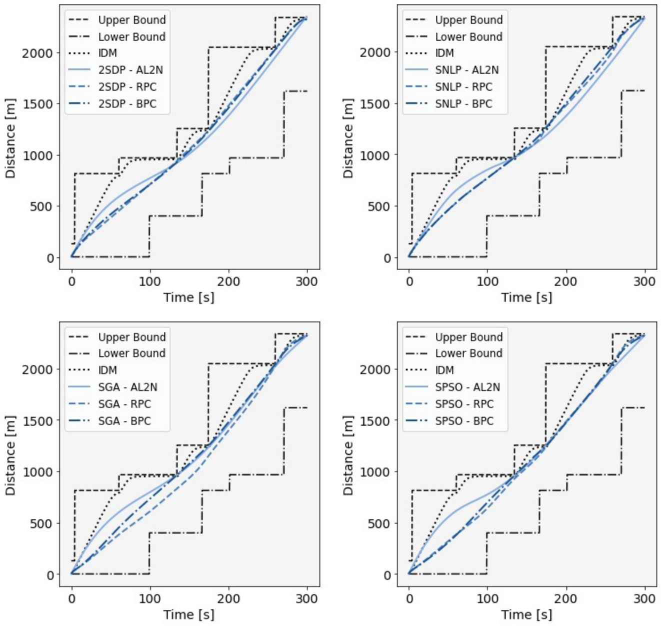


Figure 7.6 Example position versus time traces for all the methods and cost functions

Non-optimal methods may also be used to generate eco-driving traces. Four optimal methods for generating optimal eco-driving traces were compared with IDM where IDM parameters used were chosen to be representative of normal driving behavior. IDM parameters can also be chosen to result in improved EE. The IDM acceleration parameters and aggression parameters were shown to have large and significant effects on EE in Section 5. By reducing the allowed accelerations by a factor of 10–0.5 m/s^2, a mean EE improvement of around 5% was attained. An EE improvement of 5% is at the low end of what was attained with the optimal methods. IDM is a low-cost algorithm, which requires no look-ahead information, making it much easier to implement than the optimal methods. There are, however, advantages to the optimal methods over non-optimal methods regardless of how the non-optimal methods are applied. Optimal control allows for a degree of performance and flexibility that non-optimal control does not. The boundary conditions used for all the optimal methods used in this study required that the vehicles arrive at a given distance at a given time, thus maintaining a precise average speed. The solvers were able to still produce significantly more efficient traces than baseline. Low acceleration IDM, in contrast, cannot meet the same final condition and was able to improve over baseline principally by traveling at a lower mean speed. In generating optimal eco-driving traces, there is a balance between maximizing EE and limiting travel time. In this study, the travel-time aspect was removed from consideration by applying a strict final condition, but all the optimal control methods presented could be modified to allow for a precise tradeoff between EE and travel time. Thus, while requiring more in terms of increased computational load and look-ahead information, optimal control does enable more precision and flexibility than non-optimal control.

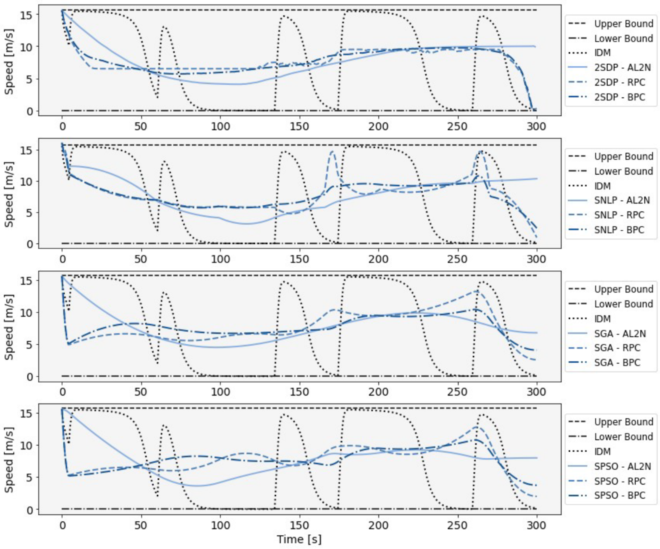


Figure 7.7 Example velocity versus time traces for all the methods and cost functions

7.3 Summary

The mean results for the methods and cost functions are presented for run-time and EE improvement in Figure 7.8. Several observations can be made. The first is that the globally optimal solution produced by 2SDP is usually significantly more efficient than the locally optimal solutions but requires much more run-time. DP can be made to run quicker with partial parallelization [91], but this would not be enough to reduce the run-times to the level of the PTO methods.

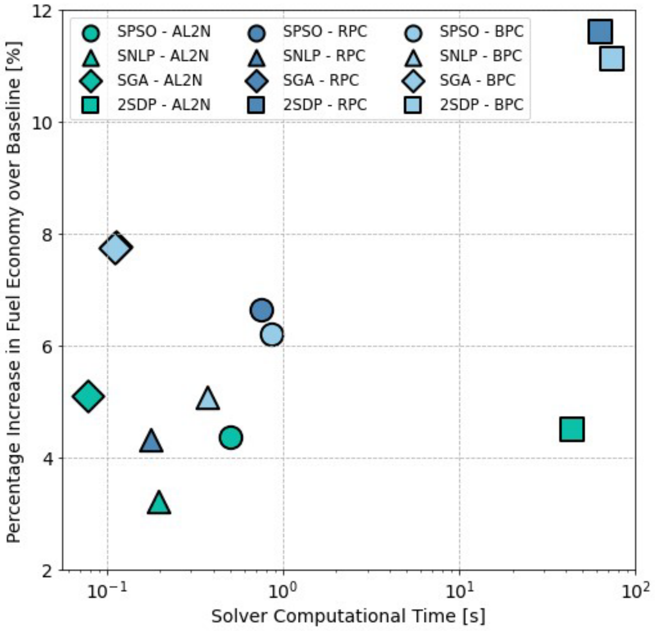


Figure 7.8 Comparison of run-time and EE improvement means for all the methods and cost functions

The literature contains several papers describing the practical implementation of multistate DP-based eco-driving control algorithms, but these either evaluate the problem on a less than 1-Hz basis or use sub-optimal approximations of the cost-to-go function. The DP-based optimal eco-driving trace solvers seem unlikely to become the basis of widely available commercial eco-driving systems due to the computational cost unless they rely on cloud computing.

Of the PTO solvers implemented, it is clear that the GA-based method was most effective in both criteria of evaluation. Visually, the SGA method with the RPC and BPC cost functions occupies a position up and to the right of the general trend line seen in Figure 7.8, indicating a favorable performance in both criteria. One reason for this is that the specifics of the optimal eco-driving trace generation problem, as defined in this study, lend to the strengths of the GA, which can explore complex optimization spaces quickly and efficiently by exploring many directions simultaneously and removing poor solutions from the selection pool. PSO also explores many solutions simultaneously, but those particles, which are seeded in low reward regions, have to gradually approach better solutions.

The specifics of the optimal eco-driving trace problem as posed in this study did not favor the SNLP or SPSO methods. At their cores, IPOPT and PSO are gradient search algorithms, extensions of Newton's method, and require the computation of a gradient at each optimization step. With nonlinearity caused by the interpolation polynomials and the nonconvexity of the constraints, such gradient search methods were less effective. It is not surprising that the SGA method was found to be the best of the type.

Another observation from Figure 7.8 is the benefit of additional information to solvers in generating the optimal eco-driving trace. In the literature, it is common to see minimization of acceleration used as a proxy for maximization of EE. This trend of acceleration minimization is also very common in robotics control literature, and as many concepts in CAV control arise from robotics it is easy to see the origins of the assumption. The assumption that acceleration minimization is a valid proxy for EE optimization is stated explicitly in several well-cited papers [74], [75]. A primary reason for the use of the l^2 norm of acceleration for a cost function is that it is independent of vehicle and powertrain parameters, and the optimization can be reduced to a conventional quadratic programming problem. The outcomes of this study indicate that cost functions, which incorporated more information about the vehicle such as velocity, aerodynamic characteristics, rolling resistance, and powertrain efficiencies, enabled optimizers to achieve higher EE for electric vehicles assuming perfect preview of the constraints. Note that the velocity-sensitive cost functions in this study require minimal additional time to compute compared with the l^2 norm of acceleration.

8. CONCLUSION

A great breadth of knowledge on the subject of autonomous eco-driving control has been generated by the research community in recent years. As the vehicular and infrastructural technology, which enables vehicular autonomous control to become more widespread, the opportunity to apply this knowledge in production vehicles becomes more achievable. A comprehensive, implementation-oriented analysis was performed to compare the relative merits of several optimization methods found in the literature. A survey of the literature was conducted, and four representative optimization methods (2SDP, and trajectory optimization with IPOPT GA, and PSO) were implemented and refined for application in simulation with real-world infrastructure data. Numerical simulations were then conducted on these methods using three progressively less abstracted cost functions (l^2 norm of acceleration, road power, and calculated battery power), and each was evaluated relative to the others in terms of performance and run-time. From these simulations, the following conclusions were reached:

1) Minimizing the l^2 norm of acceleration is confirmed to provide EE improvements.
2) Speed-sensitive cost functions that reflect vehicle and powertrain characteristics can yield improved EE results over the l^2 norm of acceleration for electric vehicles.
3) DP methods offer the highest potential for EE improvement (in the range of 7%–15%) but are extremely computationally expensive compared with other methods requiring on the order of 100×–1,000× as long to execute.
4) GA showed the most potential as a real-time method based on its relatively high performance (5%–10% EE improvement) in EE optimization and its low computational cost.

Near-future ADAS systems are anticipated to include high performance in-vehicle computers and/or embedded hardware, which is capable of computing and executing eco-driving control, meaning that this technology can be implemented as a software update. If a significant proportion of vehicles use this technology, the national energy savings could be significant.

From the selected optimization approaches considered, the results suggest the use of a GA method with an RPC cost function as providing the best tradeoff between achievable EE and computational overhead for optimal eco-driving trace generation for urban eco-driving BEVs.

9. REFERENCES

[1] E. F. Choma et al., “Health benefits of decreases in on-road transportation emissions in the United States from 2008 to 2017,” Proceedings of the National Academy of Sciences USA, vol. 118, no. 51, Dec. 2021, Art. no. e2107402118.

[2] D. L. Greene, C. B. Sims, and M. Muratori, “Two trillion gallons: Fuel savings from fuel economy improvements to U.S. light-duty vehicles, 1975–2018,” *Energy Policy*, vol. 142, Jul. 2020, Art. no. 111517.

[3] C. A. Miller, “Fifty years of EPA science for air quality management and control,” Environmental Management, vol. 67, no. 6, pp. 1017–1028, Jun. 2021.

[4] P. Slowik and N. Lutsey. *The Continued Transition to Electric Vehicles in U.S. Cities*. Accessed: Nov. 16, 2021. [Online].

Available: https://theicct.org/wp-content/uploads/2021/06/Transition_ EV_US_Cities_20180724.pdf

[5] L. Spangher, W. Gorman, G. Bauer, Y. Xu, and C. Atkinson, “Quantifying the impact of U.S. electric vehicle sales on light-duty vehicle fleet CO_2 emissions using a novel agent-based simulation,” *Transportation Research Part D: Transport and Environment*, vol. 72, pp. 358–377, Jul. 2019.

[6] S. Javanmardi, E. Bideaux, J. F. Trégouët, R. Trigui, H. Tattegrain, and E. N. Bourles, “Driving style modelling for eco-driving applications,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 13866–13871, Jul. 2017.

[7] S.-H. Ho, Y.-D. Wong, and V. W.-C. Chang, “What can eco-driving do for sustainable road transport? Perspectives from a city (Singapore) ecodriving programme,” *Sustainable Cities and Society*, vol. 14, pp. 82–88, Feb. 2015.

[8] F. M. Araghi et al., *Machine Learning and Optimization Techniques for Automotive Cyber-Physical Systems*. Berlin, Germany: Springer Nature, 2022.

[9] A. S. M. Bakibillah, M. A. S. Kamal, and C. P. Tan, “Sustainable ecodriving strategy at signalized intersections from driving data,” in *Proceedings of 59th Annual Conference of the Society of Instrument and Control Engineers of Japan. (SICE)*, Sep. 2020, pp. 165–170.

[10] A. Kesting, M. Treiber, M. Schönhof, and D. Helbing, “Adaptive cruise control design for active congestion avoidance,” *Transportation Research Part C: Emerging Technologies,* vol. 16, no. 6, pp. 668–683, Dec. 2008.

[11] M. Treiber, A. Hennecke, and D. Helbing, “Congested traffic states in empirical observations and microscopic simulations,” *Physical review. E, Statistical physics, plasmas, fluids, and related interdisciplinary topics*, vol. 62, no. 2, pp. 1805–1824, Aug. 2000, doi: 10.1103/physreve.62.1805.

[12] C. Lu, J. Dong, L. Hu, and C. Liu, “An ecological adaptive cruise control for mixed traffic and its stabilization effect,” *IEEE Access*, vol. 7, pp. 81246–81256, 2019.

[13] Q. Xin, R. Fu, W. Yuan, Q. Liu, and S. Yu, “Predictive intelligent driver model for eco-driving using upcoming traffic signal information,” *Physica A: Statistical Mechanics and its Applications*, vol. 508, pp. 806–823, Oct. 2018.

[14] M. Zhou, X. Qu, and S. Jin, “On the impact of cooperative autonomous vehicles in improving freeway merging: A modified intelligent driver model-based approach,” *IEEE Transactions on Intelligent Transportation Systems,* vol. 18, no. 6, pp. 1422–1428, Jun. 2017.

[15] Z. Li, W. Li, S. Xu, and Y. Qian, “Stability analysis of an extended intelligent driver model and its simulations under open boundary condition,” *Physica A: Statistical Mechanics and its Applications*, vol. 419, pp. 526–536, Feb. 2015.

[16] H. Rakha and R. K. Kamalanathsharma, “Eco-driving at signalized intersections using V2I communication,” in *Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Oct. 2011, pp. 341–346.

[17] Z. Nie and H. Farzaneh, “Role of model predictive control for enhancing eco-driving of electric vehicles in urban transport system of Japan,” *Sustainability*, vol. 13, no. 16, p. 9173, Aug. 2021.

[18] W. Chen, Y. Liu, X. Yang, Y. Bai, Y. Gao, and P. Li, “Platoon-based speed control algorithm for ecodriving at signalized intersection,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2489, no. 1, pp. 29–38, Jan. 2015.

[19] C. Lu and A. Aakre, “A new adaptive cruise control strategy and its stabilization effect on traffic flow,” *European Transport Research Review,* vol. 10, no. 2, p. 49, Jun. 2018.

[20] R. Bellman, “Dynamic programming and Lagrange multipliers,” *Proceedings of the National Academy of Science USA*, vol. 42, no. 10, pp. 767–769, Oct. 1956.

[21] D. E. Kirk, *Optimal Control Theory: An Introduction*. Upper Saddle River, NJ, USA: Prentice-Hall, 1970.

[22] M. Miyatake, M. Kuriyama, and Y. Takeda, “Theoretical study on ecodriving technique for an electric vehicle considering traffic signals,” in *Proceedings of the IEEE Ninth International Conference on Power Electronics and Drive Systems*, Dec. 2011, pp. 733–738.

[23] M. Kuriyama, S. Yamamoto, and M. Miyatake, “Theoretical study on eco-driving technique for an electric vehicle with dynamic programming,” in *Proceedings of the International Conference on Electrical Machines and Systems,* Oct. 2010, pp. 2026–2030.

[24] E. Hellström, M. Ivarsson, J. Åslund, and L. Nielsen, “Look-ahead control for heavy trucks to minimize trip time and fuel consumption,” *Control Engineering Practice*, vol. 17, no. 2, pp. 245–254, Feb. 2009.

[25] D. Maamria, K. Gillet, G. Colin, Y. Chamaillard, and C. Nouillant, “On the use of dynamic programming in eco-driving cycle computation for electric vehicles,” in *IEEE Conference on Control Applications (CCA),* Sep. 2016, pp. 1288–1293.

[26] D. Maamria, K. Gillet, G. Colin, Y. Chamaillard, and C. Nouillant, “Optimal eco-driving for conventional vehicles: Simulation and experiment,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 12557–12562, Jul. 2017.

[27] S. R. Deshpande, S. Gupta, A. Gupta, and M. Canova, “Real-time ecodriving control in electrified connected and autonomous vehicles using approximate dynamic programing,” *Journal of Dynamic Systems, Measurement, and Control,* vol. 144, no. 1, Jan. 2022, Art. no. 011111.

[28] S. Gupta, S. D’Alessandro, A. Gupta, S. Stockar, and M. Canova, “A computationally efficient algorithm for perturbed dynamic programs (A-PDP),” *IEEE Control Systems Letters*, vol. 7, pp. 847–852, 2023.

[29] A. Vahidi and A. Sciarretta, “Energy saving potentials of connected and automated vehicles,” *Transportation Research Part C: Emerging Technologies,* vol. 95, pp. 822–843, Oct. 2018.

[30] T. Stanger and L. D. Re, “A model predictive cooperative adaptive cruise control approach,” in *Proceedings of the American Control Conference (ACC),* Jun. 2013, pp. 1374–1379.

[31] S. Xu and H. Peng, “Design and comparison of fuel-saving speed planning algorithms for automated vehicles,” *IEEE Access*, vol. 6, pp. 9070–9080, 2018.

[32] B. Groelke, J. Borek, C. Earnhardt, J. Li, S. Geyer, and C. Vermillion, “A comparative assessment of economic model predictive control strategies for fuel economy optimization of heavy-duty trucks,” in *Proceedings of the American Control Conference (ACC),* Jun. 2018, pp. 834–839.

[33] S. Bae, Y. Choi, Y. Kim, J. Guanetti, F. Borrelli, and S. Moura, “Realtime ecological velocity planning for plug-in hybrid vehicles with partial communication to traffic lights,” in *Proceedings of the IEEE 58th Conference on Decision and Control (CDC)*, Dec. 2019, pp. 1279–1285.

[34] C. Sun, J. Guanetti, F. Borrelli, and S. J. Moura, “Optimal ecodriving control of connected and autonomous vehicles through signalized intersections,” *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3759–3773, May 2020.

[35] S. Bae, Y. Kim, J. Guanetti, F. Borrelli, and S. Moura, “Design and implementation of ecological adaptive cruise control for autonomous driving with communication to traffic lights,” in *Proceedings of the American Control Conference (ACC)*, Jul. 2019, pp. 4628–4634.

[36] H. Lee, N. Kim, and S. W. Cha, “Model-based reinforcement learning for eco-driving control of electric vehicles,” *IEEE Access*, vol. 8, pp. 202886–202896, 2020.

[37] S. Shalev-Shwartz, S. Shammah, and A. Shashua, “Safe, multiagent, reinforcement learning for autonomous driving,” 2016, *arXiv:1610.03295*.

[38] M. Zhu, Y. Wang, Z. Pu, J. Hu, X. Wang, and R. Ke, “Safe, efficient, and comfortable velocity control based on reinforcement learning for autonomous driving,” *Transportation Research Part C: Emerging Technologies*, vol. 117, Aug. 2020, Art. no. 102662.

[39] D. Pardo, L. Möller, M. Neunert, A. W. Winkler, and J. Buchli, “Evaluating direct transcription and nonlinear optimization methods for robot motion planning,” *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 946–953, Jul. 2016.

[40] A. Hamednia, N. K. Sharma, N. Murgovski, and J. Fredriksson, “Computationally efficient algorithm for eco-driving over long lookahead horizons,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6556–6570, Jul. 2022.

[41] Z. Khalik, G. P. Padilla, T. C. J. Romijn, and M. C. F. Donkers, “Vehicle energy management with ecodriving: A sequential quadratic programming approach with dual decomposition,” in *Proceedings of the American Control Conference (ACC)*, Jun. 2018, pp. 4002–4007.

[42] G. P. Padilla, S. Weiland, and M. C. F. Donkers, “A global optimal solution to the eco-driving problem,” *IEEE Control Systems Letters,* vol. 2, no. 4, pp. 599–604, Oct. 2018.

[43] L. V. Pérez and E. A. Pilotta, “Optimal power split in a hybrid electric vehicle using direct transcription of an optimal control problem,” *Mathematics and Computers in Simulation,* vol. 79, no. 6, pp. 1959–1970, Feb. 2009.

[44] Y. H. Mahmoud et al., “Autonomous eco-driving with traffic light and lead vehicle constraints: An application of best constrained interpolation,” *IFAC-PapersOnLine*, vol. 54, no. 10, pp. 45–50, 2021.

[45] A. L. Dontchev and I. V. Kolmanovsky, “State constraints in the linear regulator problem: A case study,” in *Proceedings of the 33rd IEEE Conference on Decision and Control,* Dec. 1994, vol. 1, pp. 949–950.

[46] A. L. Dontchev and I. Kolmanovsky, “Best interpolation in a strip II: Reduction to unconstrained convex optimization,” *Computational Optimization and Applications*, vol. 5, no. 3, pp. 233–251, May 1996.

[47] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science,* Oct. 1995, pp. 39–43.

[48] A. Turing, “Computing machinery and intelligence,” in *The Essential Turing*. Oxford, U.K.: Oxford Univ. Press, 2004.

[49] A. Fraser, *Computer Models in Genetics*. New York, NY, USA: McGraw-Hill, 1970.

[50] J. L. Crosby, *Computer Simulation in Genetics*. Hoboken, NJ, USA: Wiley, 1973.

[51] W. Jang, D. Jong, and D. Lee, "Methodology to improve driving habits by optimizing the in-vehicle data extracted from OBDII using genetic algorithm," in *Proceedings of the IEEE International Conference on Big Data and Smart Computing (BigComp)*, Jan. 2016, pp. 313–316.

[52] J. Li, M. Dridi, and A. El-Moudni, "A cooperative traffic control for the vehicles in the intersection based on the genetic algorithm," in *Proceedings of the 4th IEEE International Colloquium on Information Science and Technology (CiSt)* Oct. 2016, pp. 627–632.

[53] S. S. Sankar, Y. Xia, J. Carmai, and S. Koetniyom, "Optimal eco-driving cycles for conventional vehicles using a genetic algorithm," *Energies*, vol. 13, no. 17, p. 4362, Aug. 2020.

[54] M. Gautam, N. Bhusal, M. Benidris, and P. Fajri, "A GA-based approach to eco-driving of electric vehicles considering regenerative braking," in *Proceedings of the IEEE Conference on Technologies for Sustainability (SusTech)*, Apr. 2020, pp. 1–6.

[55] S. Torabi, M. Bellone, and M. Wahde, "Energy minimization for an electric bus using a genetic algorithm," *European Transport Research Review,* vol. 12, no. 1, p. 12, Dec. 2020.

[56] S. Kachroudi, M. Grossard, and N. Abroug, "Predictive driving guidance of full electric vehicles using particle swarm optimization," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 9, pp. 3909–3919, Nov. 2012.

[57] A. Fernández-Rodríguez, A. Fernández-Cardador, and A. P. Cucala, "Real time eco-driving of high speed trains by simulation-based dynamic multi-objective optimization," *Simulation Modelling Practice and Theory*, vol. 84, pp. 50–68, May 2018.

[58] V. Calderaro, V. Galdi, G. Graber, and A. Piccolo, "Deterministic vs heuristic algorithms for eco-driving application in metro network," in *Proceedings of the International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles (ESARS)*, Mar. 2015, pp. 1–6.

[59] B. Liu and A. El Kamel, "V2X-based decentralized cooperative adaptive cruise control in the vicinity of intersections," *IEEE Transactions on Intelligent Transportation Systems,* vol. 17, no. 3, pp. 644–658, Mar. 2016.

[60] M. Campbell, M. Egerstedt, J. P. How, and R. M. Murray, "Autonomous driving in urban environments: Approaches, lessons and challenges," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences,* vol. 368, no. 1928, pp. 4649–4672, Oct. 2010.

[61] Z. D. Asher, A. A. Patil, V. T. Wifvat, A. A. Frank, S. Samuelsen, and T. H. Bradley, "Identification and review of the research gaps preventing a realization of optimal energy management strategies in vehicles," *SAE International Journal of Alternative Powertrains*, vol. 8, no. 2, pp. 133–149, Nov. 2019.

[62] IIHS Announcement on AEB. (Dec. 2017). [Online]. Available: https://www.nhtsa.gov/press-releases/nhtsa-iihs-announcement-aeb

[63] *Dedicated Short Range Communications (DSRC) Message Set Dictionary*, SAE International Standard J2735_200911, Nov. 19, 2009. Accessed: Jun. 14, 2021. [Online]. Available: https://www.sae. org/standards/content/j2735_200911/

[64] A. Rabinowitz, F.M. Araghi, T. Gaikwad, Z.D. Asher, and T. H. Bradley, "Development and evaluation of velocity predictive optimal energy management strategies in intelligent and connected hybrid electric vehicles," *Energies*, vol. 14, no. 18, p. 5713, Sep. 2021.

[65] S. Edelstein. (Feb. 2022). "Tesla Vehicles Recalled for Rolling Past Stop Signs." [Online]. Available: https://www.greencarreports. com/news/1134940_tesla-vehicles-recalled-for-rolling-past-stop-signs

[66] A. I. Rabinowitz, T. Gaikwad, S. White, T. Bradley, and Z. Asher, "Infrastructure data streams for automotive machine learning algorithms research," SAE Tech. Paper 2020-01-0736, 2020.

[67] A. Brooker, J. Gonder, L. Wang, E. Wood, S. Lopp, and L. Ramroth, “FASTSim: A model to estimate vehicle efficiency, cost and performance,” SAE Tech. Paper 2015-01-0973, 2015.

[68] F. Viti, S. P. Hoogendoorn, H. J. van Zuylen, I. R. Wilmink, and B. van Arem, “Speed and acceleration distributions at a traffic signal analyzed from microscopic real and simulated data,” in *Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems*, Oct. 2008, pp. 651–656.

[69] R. Liu, X. Zhao, X. Zhu, and J. Ma, “Statistical characteristics of driver acceleration behaviour and its probability model,” *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 236, nos. 2–3, pp. 395–406, Feb. 2022.

[70] W. J. Schakel, V. L. Knoop, and B. van Arem, “Integrated lane change model with relaxation and synchronization,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2316, no. 1, pp. 47–57, Jan. 2012.

[71] M. Treiber and D. Helbing, “Microsimulations of freeway traffic including control measures,” *Auto*, vol. 49, no. 11, p. 478, Nov. 2001.

[72] C. Thiemann, M. Treiber, and A. Kesting, “Estimating acceleration and lane-changing dynamics from next generation simulation trajectory data,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2088, no. 1, pp. 90–101, Jan. 2008.

[73] B. Hegde, M. O’Keefe, S. Muldoon, J. Gonder, and C.-F. Chang. (Apr. 2021). “Real-World Driving Features for Identifying Intelligent Driver Model Parameters.” [Online]. Available: https://www.sae.org/publications/technical-papers/content/2021-01-0436/

[74] N. Prakash, G. Cimini, A. G. Stefanopoulou, and M. J. Brusstar, “Assessing fuel economy from automated driving: Influence of preview and velocity constraints,” in *Proceedings of the ASME Dynamic Systems and Control Conference*, 2016, pp. 1–6.

[75] N. Prakash, A. G. Stefanopoulou, A. J. Moskalik, and M. J. Brusstar, “Use of the hypothetical lead (HL) vehicle trace: A new method for evaluating fuel consumption in automated driving,” in *Proceedings of the American Control Conference (ACC)*, Jul. 2016, pp. 3486–3491.

[76] S. Nazari, N. Prakash, J. Siegel, and A. Stefanopoulou, “On the effectiveness of hybridization paired with eco-driving,” in *Proceedings of the American Control Conference (ACC)*, Jul. 2019, pp. 4635–4640.

[77] *Road Load Measurement and Dynamometer Simulation Using Coastdown Techniques*. Accessed: Jun. 14, 2021. [Online]. Available:

https://saemobilus.sae.org/content/J1263_201003

[78] J. Seok, Y. Wang, D. Filev, I. Kolmanovsky, and A. Girard, “Energy efficient control approach for automated HEV and BEV with short horizon preview information,” in *Proceedings of the Dynamic Systems and Control Conference,* Sep. 2018, pp. 1–9.

[79] J. T. Betts, “Survey of numerical methods for trajectory optimization,” *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, Mar. 1998.

[80] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2 ed. Cham, Switzerland: Springer, 2013.

[81] D. R. Herber, J. T. Allison, Y. Baryshnikov, C. Beck, and H. Kim, “Advances in combined architecture, plant, and control design,” PhD thesis, Univ. Illinois, Urbana-Champaign, IL, USA, 2017.

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

[83] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, “An Interior-Point Method for Large-Scale ℓ_1-Regularized Least Squares,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 606–617, Dec. 2007.

[84] *Downloadable Dynamometer Database: Argonne National Laboratory*. Accessed: Feb. 23, 2021. [Online]. Available: https://www.anl.gov/ es/downloadable-dynamometer-database

[85] C. Baker, M. Moniot, A. Brooker, L. Wang, E. Wood, and J. Gonder, “Future automotive systems technology simulator (FASTSim) validation report—2021,” National Renewable Energy Lab, Golden, CO, USA, Tech. Rep. NREL/TP-5400-81097, 2021.

[86] (2015). *Kia Soul EV-Specifications*. [Online]. Available: https://www. evspecifications.com/en/model/f45094

[87] F. Motallebiaraghi et al., “Autonomous eco-driving evaluation of an electric vehicle on a chassis dynamometer,” in *Proceedings of the WCX 2025 World Congress Experience,* Apr. 2023, pp. 1–11.

[88] X. Cai, H. P. Langtangen, and H. Moe, “On the performance of the Python programming language for serial and parallel scientific computations,” *Scientific Programming*, vol. 13, no. 1, pp. 31–56, 2005.

[89] G. Zaccone, *Python Parallel Programming Cookbook*. Birmingham, U.K.: Packt Publishing Limited, 2015.

[90] A. C. Serban, E. Poll, and J. Visser, “A standard driven software architecture for fully autonomous vehicles,” in *Proceedings of the IEEE International Conference on Software Architecture Companion (ICSA-C)*, Apr. 2018, pp. 120–127.

[91] I. V. Kolmanovsky, J. Sun, and S. N. Sivashankar, “An integrated software environment for powertrain feasibility assessment using optimization and optimal control,” *Asian Journal of Control*, vol. 8, no. 3, pp. 199–209, Oct. 2008.