2023-01-0679 Published 11 Apr 2023

An Ultra-Light Heuristic Algorithm for Autonomous Optimal Eco-Driving

Aaron I. Rabinowitz Colorado State University

Farhang Motallebiaraghi, Rick Meyer, and Zachary Asher Western Michigan University

Ilya Kolmanovsky University of Michigan

Thomas Bradley Colorado State University

Citation: Rabinowitz, A.I., Motallebiaraghi, F., Meyer, R., Asher, Z. et al., "An Ultra-Light Heuristic Algorithm for Autonomous Optimal Eco-Driving," SAE Technical Paper 2023-01-0679, 2023, doi:10.4271/2023-01-0679.

Received: 25 Oct 2022 Revised: 09 Jan 2023 Accepted: 05 Feb 2023

Abstract

Connected autonomy brings with it the means of significantly increasing vehicle Energy Economy (EE) through optimal Eco-Driving control. Much research has been conducted in the area of autonomous Eco-Driving control via various methods. Generally, proposed algorithms fall into the broad categories of rules-based controls, optimal controls, and meta-heuristics. Proposed algorithms also vary in cost function type with the 2-norm of acceleration being common. In a previous study the authors classified and implemented commonly represented methods from the literature using real-world data. Results from the study showed a tradeoff between EE improvement and run-time and that the best overall performers were meta-heuristics. Results also showed that cost functions sensitive to the 1-norm of acceleration led to better performance than those which directly minimize the 2-norm. In this paper the authors present an ultra-light heuristic method for generating optimal Eco-Driving traces for Connected Autonomous Vehicles (CAVs) which indirectly minimizes the 1-norm of acceleration. This novel method produces EE improvements in line with previously implemented meta-heuristic methods while executing in a fraction of the time.

Introduction

Optimal Eco-Driving control for CAVs presents an opportunity to leverage modern and near-future technology to improve individual vehicle and fleet efficiency. Eco-Driving, which is a strategy designed to reduce fuel consumption by minimizing accelerations and unnecessary braking events has been well known and has been shown to be effective when employed by human drivers [1]. Compared to human drivers, autonomous vehicles possess several advantages relevant to Eco-Driving. CAVs are able to precisely follow optimal driving traces, account for traffic information which is beyond line-of-sight, and do not require training time in order to implement new controls. Thus it is logical that Eco-Driving control on CAVs would produce significant efficiency improvements.

More than 10% of new passenger vehicles sold in the US currently are SAE level 2 autonomous or greater [2, 3] and the trend towards greater vehicle autonomy and greater reliance on vehicle longitudinal autonomy for normal driving is expected to increase in coming years. Similarly, the prevalence of communications infrastructure which will enable Vehicle to Everything (V2X) communication via the SAE J2735 protocol [4] should increase over the same time frame. Autonomous vehicles can simply replicate human Eco-Driving heuristics such as reducing maximum accelerations, maintaining longer follow distances, coasting towards red lights, and delaying braking but this type of control comes at the cost of causing individual vehicles to travel slower and take longer to reach their destinations. Optimal control can be used to enable vehicles to travel more efficiently while maintaining the same average speed over a given distance.

A great diversity of proposed optimal Eco-Driving algorithms exist in the literature. The reason for this diversity is the complicated nature of the problem and the many dimensional design space which results from it. In a previous study [5] the authors compared common optimal Eco-Driving trace generation methods seen in literature using a common framework and real-world traffic signal data. The methods implemented were Dynamic Programming (DP), Interior-Point Optimization (IPOPT), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO). The results of the study indicated that the best optimal Eco-Driving trace generation method

for practical implementation was GA based on the efficiency gains produced and its relatively quick run-time. This study also concerned the relative utility of several decreasingly abstract cost functions including the 2-norm of acceleration (Acceleration L^2 Norm (AL2N)), the road loads ABC equation [6] multiplied by velocity (Road Power Cost (RPC)), and a battery power equation based on the Future Automotive Systems Technology Simulator (FASTSim) calculation (Battery Power Cost (BPC)). The study found that the velocity sensitive cost functions allowed for more efficient traces to be generated than the 2-norm of acceleration cost function did for a 2015 Kia Soul Electric Vehicle (EV) model.

While performing the previously mentioned study, the authors observed that, when using velocity sensitive cost functions (RPC and BPC), the optimal solutions often resembled an attempt to "draw the straightest line possible" on a plot of distance vs. time. By contrast, when using the 2-norm of acceleration as the cost function, the solutions resembled an attempt to "draw the smoothest curve possible". This behavior could be explained in two ways. Firstly, the velocity sensitive cost functions are sensitive to velocity to the first, second, and third power but only sensitive to acceleration to the first power. Because of this, the solvers would be incentivized to lower velocity at all times even if this meant more and/or harder accelerations. Secondly, the vehicle model used in the study was an EV. Because EVs are capable of regenerating energy during decelerations the net EE penalty for accelerations was limited compared to an equivalent Internal Combustion Vehicle (ICV). These observations led to the hypothesis that an ultra-light algorithm could be developed which would mimic the optimal Eco-Driving traces generated using optimal solver methods but could execute in a fraction of the time that even the quickest optimal method could.

In this study the novel ultra-light heuristic method dubbed Indirect Net Power Minimization (INPM) is compared against best methods identified in the authors' comparative study [5] in terms of reduction of increase in EE over baseline and required computational time. A review of the current state of the literature is discussed in the Literature Review section, an overview of the optimal Eco-Driving trace generation problem and constraints is presented in the Problem Definition section, solver methods used in the study are explained in the Methods section, results are conferred in the Results section, and conclusions are given in the Conclusions section.

Literature Review

Eco-Driving Trace Generation Methods

In order to enable the practical implementation and commercialization of optimal Eco-Driving control, a method of generating optimal traces which provides credible improvements to EE and is computationally inexpensive enough to be used as part of a real-time control must be identified. Much research exists in the area of autonomous Eco-Driving controls and many methods for optimal Eco-Driving trace generation have been presented and evaluated individually. The methods presented may broadly be divided into optimal methods, metaheuristics, and rules-based methods. The optimal methods may be sub-divided into globally optimal methods, and non-globally optimal methods.

Proposed globally optimal methods are uniformly DP based. The optimal Eco-Driving trace generation problem is at least a 2-state, 1-control problem where the required states are velocity and position and the control is acceleration. A 2 state 1 control DP algorithm is presented in [7, 8] which directly optimizes a velocity trace to minimize fuel consumption while navigating around traffic signals. The primary issue with DP as an Eco-Driving control method is computational load. Using the "top-down" implementation [9] every possible combination of discrete valued states and controls must be evaluated at each time step which is challenging because the size of the problem increases exponentially with additional states and controls. The issue is commonly referred to as the "curse of dimensionality". This run-time issue is addressed by Maaria et al. in [10] where, from an initial 3 states, position, velocity, and State of Charge (SOC), the state space is ultimately reduced to just velocity. The state space reduction is accomplished in 2 ways. First, the SOC dynamics are removed from the problem which is demonstrated to have little impact on the results. Next, the position state is removed from the problem by instead formulating the optimization problem in terms of distance by using a tunable parameter-based cost to ensure that the final distance is reached at the correct time. This tunable parameter must be found for each solution via numerical root-finding. Overall this method, which can be thought of as a pseudo 2 state DP method, was implemented by the authors and found to execute in less time than 2 state 1 control DP for the same problem even though the pseudo 2 state DP algorithm had to run multiple times to find a solution. Maaria's results underline the computational cost increase associated with increasing the dimensionality of a DP problem. A second variety of DP based Eco-Driving control is proposed by Mensing *et.al.* in [11] which attempts to optimize the velocity of a vehicle following another vehicle using a 2 state 1 control DP formulation where the states are velocity and follow distance and the control is acceleration. This control is of limited practicality as it relies on future knowledge of the motion of the lead vehicle.

In order to be a real-time control any DP based method would have to be implemented in a Model Predictive Control (MPC) formulation. Many papers have proposed solver methods with DP in a MPC formulation which nominally allows DP based solvers to be used in a real-time context [12, 13, 14, 15, 16, 17, 18, 19]. Three of the cited papers presented their methods explicitly as real-time solver methods and included real-world validation studies. The first paper, [16], proposes an Approximate Dynamic Programming (ADP) solver used in an MPC formulation which is able to account for upcoming traffic signal information. The roll-out method is used to compute the cost-to-go function wherein the cost-to-go is approximated by a non-optimal policy rather than by interpolation as in a classical DP formulation. In the second paper, [17], Hellstrom et al. proposes a 2 state DP method for setting targets for the cruise control system on a semi truck

traveling on a highway without traffic signals. This method was also validated on a real vehicle and, shown feasible to be implemented in real time. Finally, Bae et al. proposed in [20] an Eco-Adaptive Cruise Control (ACC) method which uses Vehicle to Infrastructure (V2I) information and DP to set velocity targets for a cruise control system for urban driving. In this formulation, the high level logic which sets the reference velocity requires multiple seconds to compute and thus executes at a significantly slower frequency than the lower-level controller which executes it's own routine multiple times per second. It is notable that in order to utilize DP based methods as real-time controls the optimal solution has to be computed at a low frequency and the second-to-second operation of the vehicle has to be done by a cruise control system or, in the case of [16], the most computationally costly element of DP, evaluating the cost-to-go function, must be approximated by a non-optimal policy. It may also be possible to use optimal traces calculated *a priori* for given scenarios in a manner similar to the method presented in [21] for optimizing Hybrid Electric Vehicle (HEV) powertrain controls during acceleration events. In all scenarios it must be assumed that the ultimate behavior of the vehicle will not be globally optimal. One could reasonably conclude that globally optimal methods are not feasible for optimal Eco-Driving control in urban driving conditions.

There are also proposed methods for optimal controls which are not globally optimal. Two studies stand out in this space. A Model Based Reinforcement Learning (MBRL) method is proposed in [22] for optimizing motor power control for an electric vehicle subject to road grade but not traffic. The MBRL control was found to perform nearly as well as DP for the same problem. A second approach is [23] in which best interpolation splines [24, 25] and IPOPT are used to generate an optimal Eco-Driving trace in distance and time subject to lead and follow constraints. Used in an urban scenario, this method allowed for significantly greater EE to be obtained over a recorded baseline cycle.

Metaheuristics are also commonly seen in the literature. The most common of these metaheuristics are GA [26, 27, 28] and PSO [29]. In recent studies [30, 31, 32], GA methods applied to both conventional and electric vehicles showed fuel economy improvements. GA is well suited towards parallel processing which increases its potential for use in real-time controls. Although GA can be implemented onboard for real-time control using pure serial processing, the value of parallel computing for a GA method was demonstrated in [30] where significant reductions in run-time were achieved compared to serial processing. For Eco-Driving control, PSO was employed in various studies to optimize energy consumption for individual vehicles [33, 34, 35, 36] and to streamline vehicle platoon behavior at intersections [37]. A comparison of PSO with DP [36] found that PSO significantly under-performed DP in terms of EE but executed in much less time.

Rules-based methods form the last of the classifications. Rather than attempting to find a globally optimal solution for the vehicle's future velocity, rules based methods attempt to improve EE through the implementation of instantaneous control policies which limit inefficient behaviors such as accelerations, decelerations, and driving at high speeds. Such methods are relatively simple to implement while still being capable of yielding considerable fuel economy improvement [38, 39]. A common rules-based algorithm is the Intelligent Driver Model (IDM) [40] with several works presenting modified versions of the method in Eco-Driving simulations [41, 42, 43, 44]. Although non IDM rules-based methods appear in the literature [45, 46, 47], IDM and its derivatives dominate rules-based Eco-Driving literature and are often used as a baseline to compare against in optimal Eco-Driving literature.

Globally optimal and meta-heuristic methods require the evaluation of a cost function. In literature, the cost, J is evaluated either directly through an energy consumption model or indirectly via a proxy metric. The advantage of directly computing energy consumption is that it should lead to a better solution. However accurate energy consumption models can be computationally expensive. The advantage of a proxy metric is that it is usually inexpensive to compute. The correlation between the proxy metric and energy consumption may not be very strong. It is common for the proxy metric to be acceleration based. The case for using an acceleration based cost function is made in several articles [48, 49, 50]. The authors found that proxy metrics based on the Road loads ABC equation [6] correlated with increased EE far better for EVs than purely acceleration based cost functions [5].

Eco-Driving Problem Assumptions

General agreement exists in the literature that optimal Eco-Driving control is likely to be integrated into CAV control systems as an upper-level target setting algorithm which informs a lower-level target matching speed control algorithm as illustrated in Figure 1.

Such a control scheme is likely because it allows for the integration of the Eco-Driving control with existing vehicle cruise control systems which rely on distributed computing [51]. Current vehicles equipped with ACC features generally use such a configuration wherein an Advanced Driver Assistance System (ADAS) controller provides targets to the cruise control which is hosted on a separate controller. Increasing autonomy will likely result in an increase in the size and power of the ADAS computing system on vehicles but not the overall framework. Reviewed papers which featured physical validation of optimal Eco-Driving control defined the control system similarly [16, 17, 20].

FIGURE 1 Eco-Driving System Schematic

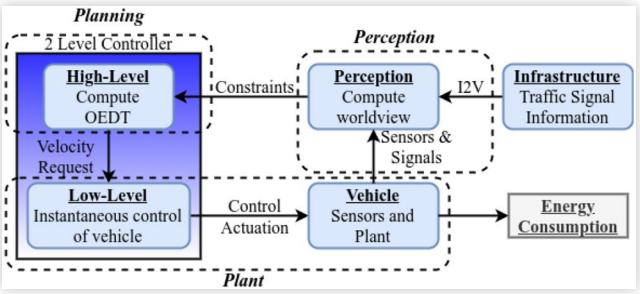


TABLE 1 Data Available to CAVs

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

Another point of broad agreement in the literature is that optimal Eco-Driving control must be applied in a receding-horizon or MPC framework [12], [13], [14], [15], [17], [18], [19]. The optimal and metaheuristic methods reviewed require knowledge or assumptions about future conditions in order to compute an Eco-Driving trace. Information commonly called for includes future traffic signal information, speed limits, and road grade. The authors contend that the information contained in Table 1 is currently, or will in the near future be, available to all CAVs [52].

Problem Definition

One of the reasons that a comparative assessment of the Eco-Driving literature is difficult is that the problem definition varies between publications. In order to make an objective comparative assessment, a common method of assessment must be employed and this necessitates a common problem definition. The purpose of optimal Eco-Driving control is to minimize energy consumption over a given distance of driving. The problem can be stated as

$$\min_{\bar{U}} J\left(S_0,\bar{U}\right) \tag{1}$$

where

$$J\left(S_0,\bar{U}\right)=\Phi\left(S_N\right)+\sum_{k=1}^{N}\Psi\left(S_k,U_k\right) \tag{2}$$

s.t.

$$S_{k+1}=f\left(S_k,U_k\right),\quad k=0,\ldots,N-1 \tag{3}$$

$$B_L\left(t\right)\le S\left(t\right)\le B_U\left(t\right) \tag{4}$$

where $\Psi\left(\bar{S},\bar{U}\right)$ is the running cost, $\Phi\left(\bar{S}\right)$ is the final state cost, $\bar{S}=\left[x,v\right]^T$ is the state vector containing the problem states position and velocity, $\bar{S_0}=\left[x_0,v_0\right]^T$ is the initial values of the state vector, $\bar{U}=\left[a\right]$ is the control vector containing the control acceleration, J is the cost (energy consumption or a proxy for energy consumption) for S and U, and B_L and B_U are vectors containing the constraints as described further in this section. The overline indicates an array containing values at multiple discrete time intervals. The goal of the optimization is to find the optimal Eco-Driving trace (U^*) such that J^* is equal to the global minimum value for J.

Major differences exist between urban Eco-Driving and rural and highway Eco-Driving. Urban driving occurs at lower speeds and involves more frequent accelerations and periods of idling due to the prominence of signalized intersections. For rural and highway driving, Eco-Driving traces are largely influenced by speed limits and road grades where for urban conditions, Eco-Driving traces are mostly effected by speed limits and traffic signals. Because urban driving presents the greater opportunity for EE improvement the authors chose to focus on urban conditions. It is important that Eco-Driving vehicles do not violate the law or exhibit extreme and unfamiliar behavior. Unlawful or extreme and unfamiliar driving behavior is likely to result in traffic accidents and additional congestion [53, 54, 55]. In order to ensure that Eco-Driving vehicles behave within the range of normal driving behavior several constraints were applied to the optimization problem.

1. The ego vehicle cannot exceed the speed limit.
2. The ego vehicle cannot run red lights.
3. The ego vehicle cannot sit through a green light phase.
4. The ego vehicle cannot travel at an abnormally low average speed.

In order to address constraints 1, 2, and 3, an upper (B_L) and lower (B_L) boundary for the state vector were implemented. If the ego vehicle is the first vehicle in a queue then an upper boundary in position and time can be generated from SPaT knowledge as an inequality constraint,

$$x\left(t\right)<B_U\left(t\right),t\in\left[0,T\right] \tag{5}$$

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. A lower bound on distance as a function of time is also defined as an inequality constraint,

$$x\left(t\right)>B_L\left(t\right),t\in\left[0,T\right] \tag{6}$$

where B_U and B_L are piece-wise functions of time based on the positions and phases of leading traffic signals. The upper bound and lower bound combine to form a ”corridor” on the distance versus time diagram in which the vehicle must travel. An example of such a corridor is shown in Figure 2.

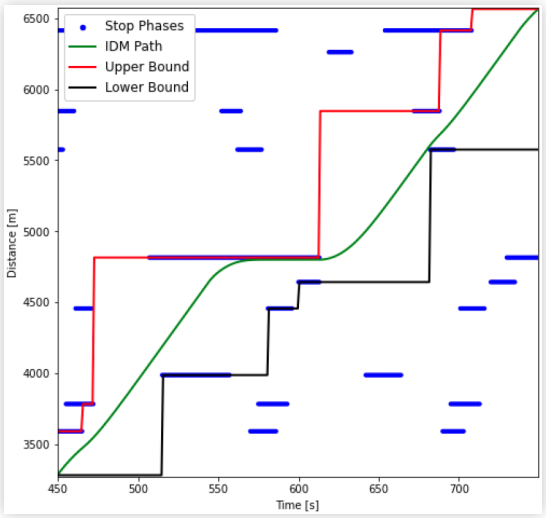
With respect to velocity, the ego vehicle velocity is required to satisfy the inequality,

$$0\le v\left(t\right)\le S_L\left(t\right),t\in\left[0,T\right] \tag{7}$$

where $S_L(t)$ is the road speed limit at time t. For this study all traffic signal and speed limit information were derived from real-world data collected in 2019 and consists of traffic light phase and timing data from 19 traffic signals along a 4 mile route in downtown Fort Collins, CO. This data was collected by the authors and its collection is described in [56].

In order to address constraint 4, a final state penalty was applied where the vehicle had to reach a given distance by the end of the time allotted. In order to ensure that the required distance, and thus average speed, reflected normal driving, a

FIGURE 2 Example upper and lower boundaries "corridor"



simulation was run using a baseline control (IDM) and all other controls were required to reach the same distance that the baseline control reached at the final time-step.

Methods

In order to evaluate the effectiveness of the novel INPM heuristic, a comparison was made with three other methods. The three methods selected were DP, GA, and IDM. DP was chosen as a comparison point due to its ability to find globally optimal solutions. GA was selected because it performed best of the selected optimal and metaheuristic methods in [5]. IDM was selected as it is a non-optimal method often used for comparison. The selected methods are explained in the subsequent subsections.

Intelligent Driver Model (IDM)

The IDM, developed by Trieber, Hennecke, and Helbing in 2000 [40] is an Rules-Based Eco-Driving (RBED) method intended to enable agent based traffic modeling. This model represented a step improvement on previous car-following models as it was meta-stable, prevented collisions, and all parameters had physical interpretations. The IDM is formulated as follows:

$$\dot{x}_i = \frac{dx_i}{dt} = v_i \tag{8}$$

$$\eta = \left(1 - \left(\frac{v_i}{v_0}\right)^{\delta} - \left(\frac{s^*\left(v_i, v_{i-1}\right)}{s_i}\right)^2\right) \tag{9}$$

$$\dot{v}_i = \frac{dv_i}{dt} = \begin{cases} a\eta & \eta \geq 0 \\ b\eta & \eta < 0 \end{cases} \tag{10}$$

$$s_i = x_{i-1} - x_i - l_{i-1} \tag{11}$$

$$s^*\left(v_i, v_{i-1}\right) = s_0 + v_i T + \frac{v_i\left(v_{i-1} - v_i\right)}{2\sqrt{ab}} \tag{12}$$

In this study 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 zero speed.

Parameter selection for IDM is important as it effects the efficiency of the generated trace. Those parameters which have 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 FASTSim [57] EV model. 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 passenger vehicle accelerations [58, 59]), 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 3.

The results of the regression analysis indicated that a, b, and δ were significant terms which negatively effected EE while none of the interaction terms were significant. Thus,

TABLE 2 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

TABLE 3 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

values for a, b, and δ can be set independently. Several papers propose methods for setting these values or the values themselves. In literature the default value for δ is given as 4 [60, 40, 61, 62] and there is a reason to hold this assumption as still valid. National Renewable Energy Laboratory (NREL) produced a report in 2021 [63] which extracted 39,000 individual driving features (acceleration-from-stop, deceleration-to-stop, and cruise events) from collected driving data and fit IDM parameters to the data. Although the IDM model used by NREL is slightly differently formulated than in this paper, the results are, nevertheless, informative. NREL found clusters for δ at .88, 1.40, 1.75, 2.13, and 4.78, ultimately the paper 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 did not see any reason to deviate from these established values for baseline control.

Since IDM parameter values can have such a major impact on EE, two IDM controls were defined. The first was Baseline IDM for which a set of normal driving parameters seen in literature were used. Baseline IDM was used as the baseline of comparison for all other methods. A Low Acceleration IDM (LAIDM) control was also defined and considered as an Eco-Driving control. For this control the maximum acceleration and decelerations were set to .5 m/s^2 or a 10 fold reduction. It should be noted that the LAIDM method cannot maintain a set average speed and thus it is not directly comparable to the other methods selected.

2 State Dynamic Programming (2SDP)

The first optimal method discussed is 2 State Dynamic Programming (2SDP). In the case of optimal Eco-Driving control, which is a 2 state 1 control non-linear optimization problem with time varying constraints, 2SDP is a natural choice and it appears in multiple forms in the literature. The dynamics of the problem in discrete-time are represented by

$$v_{k+1} = v_k + u_k \Delta t \tag{13}$$

$$x_{k+1} = x_k + v_k \Delta t \tag{14}$$

The boundary violation cost function J_{PC} is shown in equation (15), where the path constraints in x are 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.

$$J_{PC}(S_k,U_k) + \begin{cases} \beta_x (x_k - B_{L,k})^2 & x_k < B_{L,k} \\ 0 & B_{L,k} \le x_k \le B_{U,k} + \\ \beta_x (x_k - B_{U,k})^2 & x_k > B_{U,k} \end{cases} \begin{cases} \beta_v (v_k - 0)^2 & v_k < 0 \\ 0 & 0 \le v_k \le S_{L,k} \\ \beta_v (v_k - S_{L,k})^2 & v_k > S_{L,k} \end{cases} \tag{15}$$

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

$$\Phi(S_N) = \beta_{FS}(x_N - x_{target})^2 \tag{16}$$

where x_{target} is the desired ending position and β_{FS} is a weight.

SPLINE GENETIC ALGORITHM (SGA) The second optimal method discussed is the Spline Genetic Algorithm (SGA) method. The GA is employed to adjust the distances of knot points along a Piecewise Cubic Hermitic Interpolation Polynomial (PCHIP) spline in distance and time. The spline is defined in equation (17).

$$\bar{S} = PCHIP(\bar{\varepsilon}, \bar{t}_{knots}, \bar{B}_{L,knots}, \bar{B}_{U,knots}, \bar{t}) \tag{17}$$

The dynamics of the problem are represented by

$$\bar{v}_{k+1} = \bar{v}_k + \bar{u}_k \Delta t \tag{18}$$

$$\bar{x}_{k+1} = \bar{x}_k + \bar{v}_k \Delta t \tag{19}$$

The running cost and final state cost functions are the same as that for DP and are shown in equations (15) and (16) respectively.

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 used employs sorted selection wherein the best phenotypes are selected for crossover and random mutation wherein a certain percentage of the total chromosomes from all phenotypes are changed to a random number each step. The method also employs elitist carry-over wherein the best phenotype is kept for the next step un-changed. The SGA method continues to run until several termination requirements are met including a minimum number of generations and convergence of both the best and mean trace costs.

Road Power Cost (RPC) Cost Function

For both 2SDP and SGA the cost is approximated using the proxy cost function RPC. The RPC cost function is based on the road loads ABC formula [6] multiplied by velocity. This cost function takes into account the impacts of viscous and aerodynamic drag in addition to acceleration, and is given by

$$J_{RPC}(\bar{S}, \bar{U}) = \sum_{k=1}^{N} \left[A v_k + B v_k^2 + C v_k^3 + m a_k v_k \right] \tag{20}$$

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 $A = 0$, $B = C_{RR}$, 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. An approach related to RPC

has been studied in [64] under the name wheel power minimization.

Indirect Net Power Minimization (INPM)

The INPM method is designed to approximate the optimal Eco-Driving trace generated by an optimal method using a power based cost function. INPM accomplishes this by automating the process of "drawing the straightest line possible". Just as with the SGA method, the INPM works by modifying the knots of a PCHIP spline in position and time. In order to minimize run-time while ensuring that enough spline points are available to make a spline which can meet the constraints, the knot points are placed at the times of discontinuities in the upper and lower boundaries. The method is described in Algorithm 1.

INPM is a very simple and light algorithm which works by initially drawing a straight line from the start to the finish and then adjusting the positions of knots in order to avoid hitting boundaries or having to exceed the speed limit. The algorithm has two tunable parameters, the upper and lower boundary buffers which determine how close the vehicle can get to either boundary. An example of how the INPM method works is provided in Figure 3.

ALGORITHM 1 Indirect Net Power Minimization (INPM) Knot Point Generation

▷ Initializing the knots vector. All knot relative positions (ε_i) and slopes (m_i) are set to zero.

$\overline{\varepsilon} = \{0 \; \forall \; t \in \overline{t}\}$
$\varepsilon_1 = x_i$
$\varepsilon_N = x_f$
$\overline{m} = \{0 \; \forall \; t \in \overline{t}\}$
$N = length(\overline{t})$

▷ Iterating through the knots to assign distances; first and last point are pre-assigned.

for $i = 1, i = N-1, i++$ **do**

▷ If the current knot point is at a distance greater than the upper bound then the point is moved down to the upper bound. The slope is set by drawing a straight line to the last point.

  if $\varepsilon_i > B_{U,i}$ **then**
    $\varepsilon_i = B_{U,i}$
    $m = \min\left\{\left(\frac{\varepsilon_N - \varepsilon_i}{t_N - t_i}\right), S_L\right\}$

▷ If the current knot point is at a distance less than the lower bound then the point is moved up to the lower bound. The slope is set by choosing the max slope between the current knot point.

  else if $\varepsilon_i < B_{L,i}$ **then**
    $\varepsilon_i = B_{L,i}$
    $m_{max} = \min\left\{\left(\frac{\varepsilon_N - \varepsilon_i}{t_N - t_i}\right), S_L\right\}$
    for $j = i+1, j = N, j++$ **do**
      $m_{i,j} = \left(\frac{\varepsilon_j - \varepsilon_i}{t_j - t_i}\right)$
      if $m_{i,j} > m_{max}$ **then**
        $m_{max} = m_{i,j}$
      end if
    end for

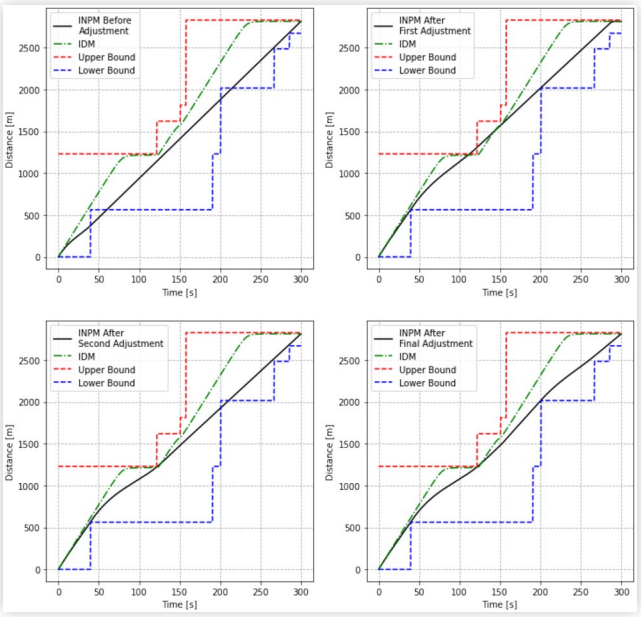
▷ If the current knot point is between the upper and lower bounds the knot is kept in place. The slope is set by choosing the max slope between the current knot point and any of the subsequent points.

  else
    $m_{max} = \min\left\{\left(\frac{\varepsilon_N - \varepsilon_i}{t_N - t_i}\right), S_L\right\}$
    for $j = i+1, j = N, j++$ **do**
      $m_{i,j} = \left(\frac{\varepsilon_j - \varepsilon_i}{t_j - t_i}\right)$
      if $m_{i,j} > m_{max}$ **then**
        $m_{max} = m_{i,j}$
      end if
    end for
  end if

▷ All subsequent knot points are recalculated based on the slope.

  for $j = i+1, j = N-1, j++$ **do**
    $\varepsilon_j = \varepsilon_i + m(t_j - t_i)$
  end for
end for

FIGURE 3 Example of Progression of INPM Solution



Vehicle Plant Model

For vehicle simulation NREL's FASTSim [57] was selected. FASTSim is an efficient, accurate, and robust Python based 1-dimensional vehicle simulation which is commonly used in research. For this study a 2017 Chevrolet Bolt EV was selected as the vehicle of interest. The 2017 Chevrolet Bolt EV is a validated FASTSim model which is provided with the program [65]. The model parameters are shown in Table 4.

Results

The INPM method was compared against the 2SDP, SGA, and LAIDM methods in terms of two criteria:

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

TABLE 4 2017 Chevrolet Bolt EV FASTSim Model Parameters

Parameter	Value	Units
Mass	1758	kg
Frontal Area	2.845	m
Coefficient of Drag (C_D)	.29	N/A
Coefficient of Rolling Resistance (C_{RR})	.0073	N/A
Maximum Battery Storage	60	kWh
Wheelbase	2.6	m
Max Motor Power	150	kW

The methods were evaluated for their ability to generate 5 minute long Eco-Driving traces. Longer time horizons allow for the solvers to improve over baseline to a greater degree but also increase the optimization space for the solvers leading to rapid growth in run-times. The 5 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 paper is only concerned with the efficacy of the INPM method compared to other methods for single evaluations.

EE Improvement

Each solver was evaluated for 100 pre-defined boundaries cases in order to evaluate the effectiveness of the solvers in terms of EE improvement. These pre-defined cases were defined by a selection of random starting times and locations on the phase map as shown in the Problem Definition section. The decision to run 100 cases per method was made in order to allow for the use of large sample statistics.

The results of the experiment in terms of EE and EE improvement over baseline are shown in Figure 4 and 5.

From Figures 4 and 5 an order in terms of relative performance for the different methods can be seen. As might be expected, the traces generated using the 2SDP method were the most efficient on average followed by the SGA generated traces. The performance of the INPM was worse, on average than that of the SGA method but better than that of the LAIDM even though the LAIDM method had the advantage of being able to reduce average speed over the drive cycle.

Because significant variance existed in the gross performance results for all studied methods as well as for the Baseline IDM method, the performance improvements showed a high degree of variances. When the boundary conditions are simpler (fewer knot points and/or wider gap between the upper and lower bounds) the optimal methods will outperform the baseline IDM to a greater degree than when the conditions are more complex. Simpler conditions allow for straighter traces to be drawn and the optimal control methods will find these low cost traces where the baseline will not. Thus the variances for EE improvement were large. In fact, even with 100 samples, the differences in performance observed between the SGA and INPM methods were statistically insignificant at 95% confidence as recorded in Figure 6.

The large variances in performance over baseline we

FIGURE 4 Mean and standard deviation of EE for all methods.

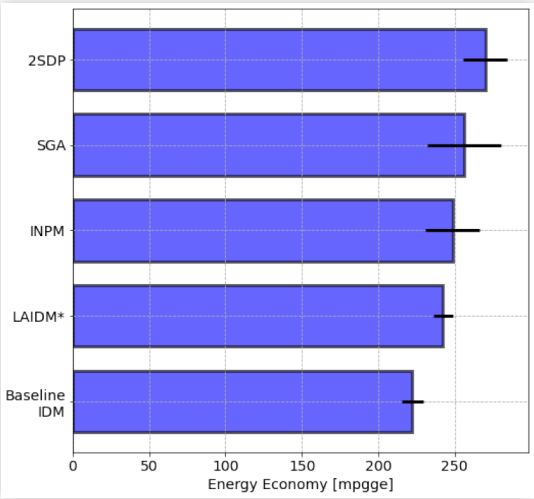


FIGURE 5 Mean and standard deviation of EE improvement over baseline results for all methods. *All methods other than LAIDM maintain a set average speed, LAIDM traces may have lower average speeds than the others

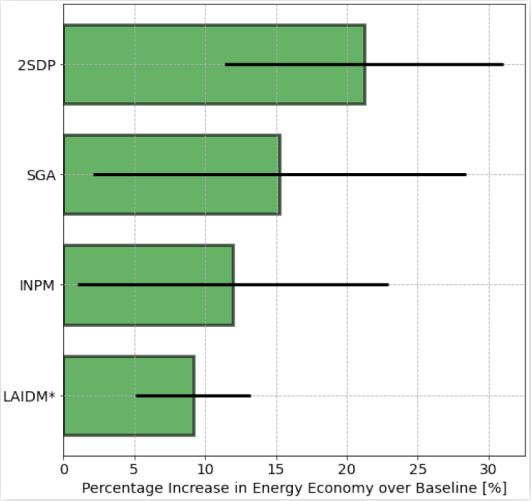


FIGURE 6 Significance of comparative results (P-values), purple indicates that the column significantly outperformed the row, blue indicates that the row significantly outperformed the column, green indicates the difference between the row and column was insignificant at 95% confidence

Method	2SDP	SGA	INPM	LAIDM
2SDP	1.0000	0.0005	0.0000	0.0000
SGA	0.0005	1.0000	0.0565	0.0000
INPM	0.0000	0.0565	1.0000	0.0214
LAIDM	0.0000	0.0000	0.0214	1.0000

Computational Load

All methods for this study were implemented in Python 3 with the NumPy and Scipy libraries. All solvers were run-time optimized in Python and all are vectorized to the highest degree possible in order to minimize run-time [66]. Nevertheless, a specific outcome of the Python implementation is that Python has very limited parallel processing capability [67] which means that the authors were not able to experiment on the impacts of parallel processing. The computer used for simulation contained an AMD Ryzen 7 3700x 8-core multi-threading capable CPU with 16 gigabytes

FIGURE 7 Log Mean and standard deviations of run-times for all methods and cost functions

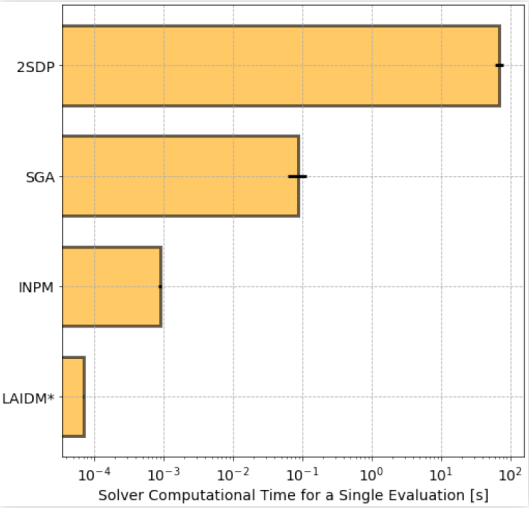
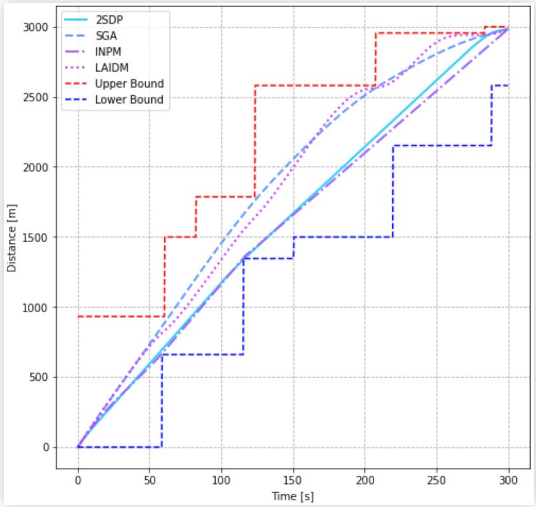


FIGURE 8 Comparison of Eco-Driving traces generated by the studied methods



of RAM and ran the 64 bit Ubuntu 18.04 LTS operating system with Python 3.8. All simulations were conducted on the same computer to ensure the integrity of relative runtimes. 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 shows the relative run-times for each method and cost function.

There were stark differences in run time required to compute a single evaluation. It should be noted that the run-times observed are properties the methods. Assuming a fair comparison DP based methods will require more computational cost to compute a solution than meta-heuristics such as GA or heuristics such as INPM and IDM. As previously mentioned, the 2SDP, SGA, and INPM methods must be used in a receding horizon context and thus at each evaluation a whole trace must be computed. For IDM, only the acceleration for the next step must be computed. This taken into account, it is impressive that the INPM solver required only about 10 times as long to compute a single evaluation as the IDM method required. It is also worth noting the practical difference in required computational time between the INPM and SGA methods. While the INPM method could execute at a rate of 1000 Hz, the SGA method could only execute at a frequency of 10 Hz. One would expect that after conversion to a compiled language and runtime optimization, the computational time required for the INPM method might become trivial even on low power computers while achieving the same for SGA would likely require computers capable of highly parallel processing.

Trace Comparison

The differences in EE improvement reflect visual differences in generated Eco-Driving traces. A representative example is shown in Figure 8 for all methods with one set of constraints.

In general, the Eco-Driving traces improve over the baseline traces primarily by minimizing the speed reduction due to traffic signals. The traces generated by the heuristic method INPM and by the rules-based method LAIDM would also be more efficient than the baseline trace and for the same reasons. There are many local optima in the results space and many are very similar to the global optimum. Thus non-DP methods such as GA are likely to settle on a local optimum. It is notable from the example trace in Figure 8 how similar the INPM trace is to the 2SDP trace with RPC as the cost function. The purpose of the INPM method was to approximate the 2SDP trace based on a set of constraints and the method works best when the two are similar.

Conclusions

The significance of the INPM method is that it provides an ultra-light method for computing Eco-Driving traces which are similar to optimal Eco-Driving traces generated by much more computationally expensive methods. In this study the novel INPM method was used to generate Eco-Driving traces for a 2017 Chevrolet Bolt EV FASTSim model using real-world traffic signal data. The EE results generated using the INPM method were compared to a DP method, a GA method, and a low acceleration IDM method.

As compared to the DP and GA methods, the INPM method is a lower performing but significantly less computationally expensive alternative. The EE improvements generated using the EE improvement method were 56% as large as those generated using the DP method and 78% as large as those generated using the GA method. Simultaneously, the INPM method executed for a single evaluation on average in 1 hundredth of the time that the GA method required and 1 hundred-thousandth of the time the DP method required. Even as implemented in Python, the INPM method was capable of executing at 1000 Hz and, because it requires a for-loop, there is reason to suspect that the method could execute at even higher frequencies if converted to a compiled language. As mentioned earlier the results of the low

acceleration IDM method were not directly comparable to those of the other methods as the IDM method was not capable of maintaining a given average speed. However, the INPM method still significantly outperformed the low acceleration IDM method in terms of EE improvement. The difference in run-time for a single evaluation between the INPM and IDM methods was roughly a factor of ten in favor of IDM but the difference is likely of little practical value due to both methods being able to execute at greater than 1000 Hz.

That the INPM method significantly outperformed the low acceleration IDM method provides further evidence to the notion advanced in [5] that, for EVs, minimizing accelerations is insufficient as a proxy for Eco-Driving. Because of regeneration, EVs do not suffer the same net efficiency penalties due to acceleration events as ICVs do. Where minimizing acceleration tends to lead to Eco-Driving traces which are smooth-as-possible curves, minimizing chassis power tends to lead to traces which are straight-as-possible lines with sharper accelerations. Thus, it is significant that a method which directly attempts to draw straight-as-possible lines outperformed one which directly limits acceleration even though the latter also often reduces average speed.

In a previous comparative study the authors identified GA based methods for optimal Eco-Driving trace generation as the most promising of several commonly seen alternatives due to their combination of EE improvement and low run-time. The novel INPM method was shown to represent a small reduction in EE improvement performance from the GA method implemented but a major reduction in run-time. As part of a two-level optimal Eco-Driving control for CAVs, the INPM method is a good option to consider for implementation especially if computing power is a strong limitation.

References

1. Javanmardi, S., Bideaux, E., Trégouët, J., Trigui, R. et al., "Driving Style Modelling for Eco-Driving Applications," *IFAC-PapersOnLine* 50, no. 1 (2017): 13866-13871.
2. "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," accessed June 15, 2022, https://www.sae.org/standards/content/j3016_202104/
3. "Canalys: 10features."
4. "J2735: Dedicated Short Range Communications (DSRC) Message Set Dictionary™ - SAE International," https://www.sae.org/standards/content/j2735_200911/. Accessed: 2021-6-14.
5. Rabinowitz, A., Ang, C.C., Mahmoud, Y.H., Motallebi Araghi, F. et al., "Real Time Implementation Comparison of Urban Eco-Driving Controls," *IEEE Transactions on Control Systems Technology* (2022) (In Review).
6. "Road Load Measurement and Dynamometer Simulation Using Coastdown Techniques," accessed June 14, 2021, https://saemobilus.sae.org/content/J1263_201003
7. Miyatake, M., Kuriyama, M., and Takeda, Y., "Theoretical Study on Eco-Driving Technique for an Electric Vehicle Considering Traffic Signals," in *2011 IEEE Ninth International Conference on Power Electronics and Drive Systems*, 733-738, 2011.
8. Kuriyama, M., Yamamoto, S., and Miyatake, M., "Theoretical Study on Eco-Driving Technique for an Electric Vehicle with Dynamic Programming," *Journal of international Conference on Electrical Machines and Systems* 1, no. 1 (2012): 114-120.
9. Kirk, D.E., *Optimal Control Theory: An Introduction* (Prentice-Hall, 1970)
10. Maamria, D., Gillet, K., Colin, G., Chamaillard, Y. et al., "On the Use of Dynamic Programming in Eco-Driving Cycle Computation for Electric Vehicles," in *2016 IEEE Conference on Control Applications (CCA)*, 1288-1293, 2016.
11. Mensing, F., Bideaux, E., Trigui, R., and Tattegrain, H., "Trajectory Optimization for Eco-Driving Taking Into Account Traffic Constraints," *Transportation Research Part D: Transport and Environment* 18 (2013): 55-61.
12. Vahidi, A. and Sciarretta, A., "Energy Saving Potentials of Connected and Automated Vehicles," *Transportation Research Part C: Emerging Technologies* 95 (2018): 822-843.
13. Stanger, T. and del Re, L., "A Model Predictive Cooperative Adaptive Cruise Control Approach," in *2013 American Control Conference*, 1374-1379, 2013.
14. Xu, S. and Peng, H., "Design and Comparison of Fuel-Saving Speed Planning Algorithms for Automated Vehicles," *IEEE Access* 6 (2018): 9070-9080.
15. Groelke, B., Borek, J., Earnhardt, C., Li, J. et al., "A Comparative Assessment Of Economic Model Predictive Control Strategies for Fuel Economy Optimization of Heavy-Duty Trucks," in *2018 Annual American Control Conference (ACC)*, 834-839, 2018.
16. Deshpande, S.R., Gupta, S., Gupta, A., and Canova, M., "Real-Time Eco-Driving Control in Electrified Connected and Autonomous Vehicles Using Approximate Dynamic Programming," 144 (2022): 011111.
17. Hellström, E., Ivarsson, M., Åslund, J., and Nielsen, L., "Look-Ahead Control for Heavy Trucks to Minimize Trip Time and Fuel Consumption," *Control Engineering Practice* 17, no. 2 (2009): 245-254.
18. Bae, S., Choi, Y., Kim, Y., Guanetti, J. et al., "Real-Time Ecological Velocity Planning for Plug-In Hybrid Vehicles with Partial Communication to Traffic Lights," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019.
19. Sun, C., Guanetti, J., Borrelli, F., and Moura, S.J., "Optimal Eco-Driving Control of Connected and Autonomous Vehicles through Signalized Intersections," *IEEE Internet of Things Journal* 7, no. 5 (2020): 3759-3773.
20. Bae, S., Kim, Y., Guanetti, J., Borrelli, F. et al., "Design and Implementation of Ecological Adaptive Cruise Control for Autonomous Driving with Communication to Traffic Lights," in *2019 American Control Conference (ACC)*, 2019.
21. Trinko, D.A., Asher, Z.D., and Bradley, T.H., "Application of Pre-Computed Acceleration Event Control to Improve Fuel Economy in Hybrid Electric Vehicles," SAE Technical Paper 2018-01-0997 (2018), https://doi.org/10.4271/2018-01-0997.
22. Lee, H., Kim, N., and Cha, S.W., "Model-Based Reinforcement Learning for Eco-Driving Control of Electric Vehicles," *IEEE Access* 8 (2020): 202886-202896.

23. Mahmoud, Y.H., Brown, N.E., Araghi, F.M. et al., "Autonomous Eco-Driving with Traffic Light and Lead Vehicle Constraints: An Application of Best Constrained Interpolation," *IFAC-PapersOnLine* 10, no. 10 (2021): 45-50.
24. Dontchev, A. and Kolmanovsky, I., "State Constraints in the Linear Regulator Problem: A Case Study," in *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, 1995.
25. Dontchev, A.L. and Kolmanovsky, I., "Best Interpolation in a Strip II: Reduction to Unconstrained Convex Optimization," *Computational Optimization and Applications* 5, no. 3 (1996): 233-251.
26. Turing, A., "Computing Machinery and Intelligence (1950)," The Essential Turing, 2004.
27. Fraser, A., *Computer Models in Genetics* (McGraw-Hill, 1970)
28. Crosby, J.L., *Computer Simulation in Genetics* (Wiley, 1973)
29. Eberhart, R. and Kennedy, J., "A New Optimizer Using Particle Swarm Theory," in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 39-43, 1995.
30. Sankar, S.S., Xia, Y., Carmai, J., and Koetniyom, S., "Optimal Eco-Driving Cycles for Conventional Vehicles Using a Genetic Algorithm," *Energies* 13, no. 17 (2020).
31. Gautam, M., Bhusal, N., Benidris, M., and Fajri, P., "A GA-Based Approach to Eco-Driving of Electric Vehicles Considering Regenerative Braking," in *2021 IEEE Conference on Technologies for Sustainability (SusTech)*, 2020.
32. Torabi, S., Bellone, M., and Wahde, M., "Energy Minimization for an Electric Bus Using a Genetic Algorithm," *European Transport Research Review* 12 (2020): 1-8.
33. Li, J., Dridi, M., and El-Moudni, A., "A Cooperative Traffic Control for the Vehicles in the Intersection Based on the Genetic Algorithm," in *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*, 627-632, 2016.
34. Kachroudi, S., Grossard, M., and Abroug, N., "Predictive Driving Guidance of Full Electric Vehicles Using Particle Swarm Optimization," *IEEE Transactions on Vehicular Technology* 61, no. 9 (2012): 3909-3919.
35. Fernández-Rodríguez, A., Fernández-Cardador, A., and Cucala, A.P., "Real Time Eco-Driving of High Speed Trains by Simulation-Based Dynamic Multi-Objective Optimization," *Simulation Modelling Practice and Theory* 84 (2018): 50-68.
36. Calderaro, V., Galdi, V., Graber, G., and Piccolo, A., "Deterministic vs Heuristic Algorithms for Eco-Driving Application in Metro Network," in *2015 International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles (ESARS)*, 1-6, 2015.
37. Liu, B. and El Kamel, A., "V2x-Based Decentralized Cooperative Adaptive Cruise Control in the Vicinity of Intersections," *IEEE Transactions on Intelligent Transportation Systems* 17, no. 3 (2016): 644-658.
38. Bakibillah, A.S.M., Kamal, M.A.S., and Tan, C.P., "Sustainable Eco-Driving Strategy at Signalized Intersections from Driving Data," in *2020 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 165-170, 2020.
39. Kesting, A., Treiber, M., Schönhof, M., and Helbing, D., "Adaptive Cruise Control Design for Active Congestion Avoidance," *Transportation Research Part C: Emerging Technologies* 16, no. 6 (2008): 668-683.
40. Treiber, M., Hennecke, A., and Helbing, D., "Congested Traffic States in Empirical Observations and Microscopic Simulations," *Physical Review E*, vol. 62, p. 1805-1824, Aug 2000, https://doi.org/10.1016/j.trc.2007.12.004.
41. Lu, C., Dong, J., Hu, L., and Liu, C., "An Ecological Adaptive Cruise Control for Mixed Traffic and Its Stabilization Effect," *IEEE Access* 7 (2019): 81246-81256.
42. Xin, Q., Fu, R., Yuan, W., Liu, Q. et al., "Predictive Intelligent Driver Model for Eco-Driving Using Upcoming Traffic Signal Information," *Physica A: Statistical Mechanics and its Applications* 508 (2018): 806-823.
43. Zhou, M., Qu, X., and Jin, S., "On the Impact of Cooperative Autonomous Vehicles in Improving Freeway Merging: A Modified Intelligent Driver Model-Based Approach," *IEEE Transactions on Intelligent Transportation Systems* 18, no. 6 (2017): 1422-1428.
44. Li, Z., Li, W., Xu, S., and Qian, Y., "Stability Analysis of an Extended Intelligent Driver Model and Its Simulations under Open Boundary Condition," *Physica A: Statistical Mechanics and its Applications* 419 (2015): 526-536.
45. Rakha, H. and Kamalanathsharma, R.K., "Eco-Driving at Signalized Intersections Using v2i Communication," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 341-346, 2011.
46. Nie, Z. and Farzaneh, H., "Role of Model Predictive Control for Enhancing Eco-Driving of Electric Vehicles in Urban Transport System of Japan," *Sustainability* 13, no. 16 (2021).
47. Chen, W., Liu, Y., Yang, X., Bai, Y. et al., "Platoon-Based Speed Control Algorithm for Ecodriving at Signalized Intersection," *Transportation Research Record* 2489, no. 1 (2015): 29-38.
48. Prakash, N., Cimini, G., Stefanopoulou, A.G., and Brusstar, M.J., "Assessing Fuel Economy from Automated Driving: Influence of Preview and Velocity Constraints," in *Volume 2: Mechatronics; Mechatronics and Controls in Advanced Manufacturing; Modeling and Control of Automotive Systems and Combustion Engines; Modeling and Validation; Motion and Vibration Control Applications; Multi-Agent and Networked Systems; Path Planning and Motion Control; Robot Manipulators; Sensors and Actuators; Tracking Control Systems; Uncertain Systems and Robustness; Unmanned, Ground and Surface Robotics; Vehicle Dynamic Controls; Vehicle Dynamics and Traffic Control*, 2016.
49. Prakash, N., Stefanopoulou, A.G., Moskalik, A.J., and Brusstar, M.J., "Use of the Hypothetical Lead (HL) Vehicle Trace: A New Method for Evaluating Fuel Consumption in Automated Driving," in *2016 American Control Conference (ACC)*, 2016.
50. Nazari, S., Prakash, N., Siegel, J., and Stefanopoulou, A., "On the Effectiveness of Hybridization Paired with Eco-Driving," in *2019 American Control Conference (ACC)*, 2019.
51. Ebert, C. and Favaro, J., "Automotive Software," *IEEE Software*, vol. 34, pp. 33-39, may 2017.

52. "IIHS Announcement on AEB," December 2017, https://www.nhtsa.gov/press-releases/nhtsa-iihs-announcement-aeb
53. World Health Organization, "Global Status Report on Road Safety 2018: Summary (no. who/nmh/nvi/18.20)," 2018.
54. Singh, S., "Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey," Technical Report, 2015.
55. Ucar, S., Hoh, B., and Oguchi, K., "Differential Deviation Based Abnormal Driving Behavior Detection," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 1553-1558, 2021.
56. Rabinowitz, A.I., Gaikwad, T., White, S., Bradley, T. et al., "Infrastructure Data Streams for Automotive Machine Learning Algorithms Research," Technical Report, SAE Technical Paper, 2020.
57. Brooker, A., Gonder, J., Wang, L., Wood, E. et al., "Fastsim: A model to estimate vehicle efficiency, cost and performance," SAE Technical Paper 2015-01-0973 (2015), https://doi.org/10.4271/2015-01-0973.
58. Viti, F., Hoogendoorn, S.P., van Zuylen, H.J., Wilmink, I.R. et al., "Speed and Acceleration Distributions at a Traffic Signal Analyzed from Microscopic Real and Simulated Data," in *2008 11th International IEEE Conference on Intelligent Transportation Systems*, 651-656, 2008.
59. Liu, R., Zhao, X., Zhu, X., and Ma, J., "Statistical Characteristics of Driver Acceleration Behaviour and Its Probability Model," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 236, no. 2-3 (2021): 395-406.
60. Schakel, W.J., Knoop, V.L., and van Arem, B., "Integrated Lane Change Model with Relaxation and Synchronization," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2316, no. 1, p. 47-57, 2012.
61. Treiber, M. and Helbing, D., "Microsimulations of Freeway Traffic Including Control Measures," *Auto* 49, no. 11 (2001): 478.
62. Thiemann, C., Treiber, M., and Kesting, A., "Estimating Acceleration and Lane-Changing Dynamics from Next Generation Simulation Trajectory Data," *Transportation Research Record: Journal of the Transportation Research Board* 2088, no. 1 (2008): 90-101.
63. Hegde, B., O'Keefe, M., Muldoon, S., Gonder, J. et al., "Real-World Driving Features for Identifying Intelligent Driver Model Parameters," April 2021, https://www.sae.org/publications/technical-papers/content/2021-01-0436/
64. Seok, J., Wang, Y., Filev, D., Kolmanovsky, I. et al., "Energy-Efficient Control Approach for Automated HEV and BEV with Short-Horizon Preview Information," in *Dynamic Systems and Control Conference*, vol. 51890, V001T10A004, American Society of Mechanical Engineers, 2018.
65. Baker, C., Moniot, M., Brooker, A., Wang, L. et al., "Future Automotive Systems Technology Simulator (Fastsim) Validation Report - 2021," 2021.
66. Cai, X., Langtangen, H.P., and Moe, H., "On the Performance of the Python Programming Language for Serial and Parallel Scientific Computations," *Scientific Programming* 13, no. 1 (2005): 31-56.
67. Zaccone, G., *Python Parallel Programming Cookbook* (Packt Publishing Limited, 2015)

ISSN 0148-7191