

Legendre Pseudospectral Computation of Optimal Speed Profiles for Vehicle Eco-Driving System*

Shaobing Xu, Kun Deng, Shengbo Eben Li, Sisi Li, and Bo Cheng

Abstract— This paper presents a computational framework to solve optimal control problems (OCPs) using Legendre Pseudospectral (PS) method and its application to obtain eco-driving strategies for ground vehicles. Both control and state variables of OCPs are approximated by Lagrange interpolating polynomials at the Legendre-Gauss-Lobatto (LGL) collocation points. The OCP is converted into a nonlinear programming (NLP) problem, and numerically solved by matured optimization algorithms. To implement the PS method, we developed a computational package, called Pseudospectral Optimal control Problem Solver (POPS) in Matlab environment. Further, the POPS is applied to obtain fuel-optimized driving strategies for automated vehicles in hilly road conditions.

Index Terms—Optimal control, Legendre Pseudospectral method, eco-driving.

I. INTRODUCTION

Due to the increasing demands on improving vehicle performance, the design of optimal control strategies for vehicles attracted more and more attentions. Typical examples include trajectory optimization for automated vehicles, fuel optimized eco-driving, power management for HEVs/EVs, and optimal control for active suspensions, etc. [1], [2]. In engineering practice, efficient numerical solvers for aforementioned optimal control problems (OCPs) are extremely important for reducing workload and cost in designing advanced control strategies.

The main methods to obtain the fuel optimized eco-driving strategies are dynamic programming and direct methods [3], [4], [5], [6]. For example, Kuriyama *et al.* optimized the speed and acceleration for electric vehicles in the ramp condition using dynamic programming [3]; Kamal *et al.* used the model predictive control to obtain the eco-acceleration in up-down slopes condition, the OCPs are solved by traditional direct methods [4]. In this paper, we use and implement Legendre Pseudospectral (PS) method to obtain eco-driving strategies. Compared to the traditional direct methods to OCP, e.g. shooting method and collocation method, the pseudospectral (PS) method is more attractive due to it merits on lower sensitivity to initial guess and faster convergence to optimal solution [7]. Dynamic programming exist the drawbacks that

it suffers from the curse of dimensionality and holds low computational efficiency [8].

In 1995, Elnagar *et al.* first introduced the concept of PS method by employing Lagrange polynomials to approximate states and control variables [10]. Since 2000, Fahroo, Ross, Rao *et al.* intensively studied the PS method, and obtained many important theoretical results, e.g., convector mapping theorem, convergence, and costate estimation [11], [12], [13], [14], [15]. There are generally four types of PS variants: Gauss method, Radau method, Legendre method, Chebyshev method [11]. The basic principle of the PS methods is to approximate state and control variables through global interpolating polynomials and convert the OCP into a nonlinear programming (NLP) problem at orthogonal collocation points [9], [10], [11]. Ross *et al.* showed that the formulated NLP problems can converge to the original OCP in spectral accuracy [16]. Lloyd further obtained the formulation of spectral accuracies for different kinds of functions summarized as follows [17]. For infinitely differentiable function, the spectral accuracy is given by $O(N^{-m})$ for every $m \in \mathbb{Z}^+$, where N denotes the number of collocation points. For analytic function, the convergence will be faster at the rate of (c^N) for some constant $0 < c < 1$. The spectral accuracy increases as the smoothness of functions increases. Huntington *et al.* compared the performance of first three types of PS. The comparison indicates that Gauss and Radau methods have similar computational accuracy, and they have better co-state estimation capabilities than Legendre method [18]. In fact, these three methods have both advantages and disadvantages for different OCPs. The Legendre method has better performance for OCPs with non-free boundary conditions, while Gauss and Radau methods may not converge as indicated by [11]. For infinite-horizon OCPs, Radau method is more applicable [14] since $\tau = 1$ is a singular point after transforming the infinite horizon into finite horizon $[-1, 1]$. Today the PS method has been successfully applied to problems in control engineering, particularly in aerospace engineering. Some PS based OCP solvers are also commercially available, e.g., DIDO [19], GPOPS [20], PSOPT [21] and PROPT [22]. The DIDO is developed by the team of Ross using Legendre PS method [19]. The GPOPS is developed by the team of Rao using Gauss and Radau PS method [20]. The PSOPT is developed by the team of Becerra, which is written in C++ using direct collocation method with both global and local polynomials [21]. The PROPT is developed by *Tomlab Optimization* using Gauss and Chebyshev PS method [22].

The main contribution of this paper is to present a computational framework to obtain the fuel-optimized strategies using Legendre pseudospectral method, and to implement the method by developed a universal solver in

*Resrach was supported by NSF China (51205228), Tsinghua University Initiative Scientific Research Program (2012THZ0).

S. Eben Li is with the State Key Laboratory of Automotive Safety and Energy, Dept. of Automotive Eng., Tsinghua University, Beijing 10084, China. (Corresponding author, e-mail: lisb04@gmail.com).

S. Xu and B. Cheng are with the State Key Laboratory of Automotive Safety and Energy, Tsinghua University, Beijing 100084, China.

K. Deng is with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA.

S. Li is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA.

Matlab environment. The developed solver is called Pseudospectral Optimal control Problem Solver (POPS), which is a good option to users who pursue a free Legendre pseudospectral solver for their complex optimal control problem. In Section II, we briefly introduce the Legendre PS method as well as how to numerically compute the collocation points. In Section III, we describe the characteristics of the POPS. In Section IV, we apply the POPS to obtain the fuel-optimized driving strategies for automated vehicles.

II. PSEUDOSPECTRAL METHOD

An OCP is to find an optimal control law $\mathbf{u}(t) \in \mathbb{R}^{N_u}$ such that the system moves from one state to another with minimal performance index. The Bolza-typed OCP is given as

$$\min_{\mathbf{u}(t)} J = \phi(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} G(\mathbf{x}(t), \mathbf{u}(t), t) dt. \quad (1)$$

s.t.

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \\ \boldsymbol{\varphi}(\mathbf{x}(t_0), \mathbf{x}(t_f), t_0, t_f) &= 0, \\ \mathbf{C}_{eq}(\mathbf{x}(t), \mathbf{u}(t), t) &= 0, \\ \mathbf{C}_{ineq}(\mathbf{x}(t), \mathbf{u}(t), t) &\leq 0, \end{aligned}$$

where $t \in \mathbb{R}$ denotes the time, $\mathbf{x} \in \mathbb{R}^{N_x}$ denotes the state vector, $\mathbf{u} \in \mathbb{R}^{N_u}$ denotes the control vector, $\mathbf{f}(\cdot)$ denotes the state function, $\phi(\cdot)$ denotes the Mayer performance index, $G(\cdot)$ denotes the Lagrange performance index, $\boldsymbol{\varphi}(\cdot)$ denotes the initial and final constraints, $\mathbf{C}_{eq}(\cdot)$ and $\mathbf{C}_{ineq}(\cdot)$ denote the equality and inequality path constraints, respectively.

A. Implementation framework

In Legendre PS method, the state and control variables are approximated by Lagrange interpolating polynomials at Legendre Gauss Lobatto (LGL) points. The state space equations are enforced into equality constraints at the LGL points. The integral of cost function is calculated by the Gauss-Lobatto quadrature rule. Then the OCP is converted into the NLP problem that can be solved by available optimization solvers. The detailed framework is described below.

Step 1: Time-domain transformation

For the sake of simplicity, we first transform the time domain $[t_0, t_f]$ to the canonical interval $[-1, 1]$ by

$$\tau = (2t - t_f - t_0)/(t_f - t_0), \tau \in [-1, 1]. \quad (2)$$

Step 2: Collocation and discretization

The LGL points come from the roots of orthogonal polynomials, which help avoid the Runge phenomena [23]. Let $P_N(\tau)$ denote the Legendre polynomials of order N [18], defined as

$$P_N(\tau) = \frac{1}{2^N N!} \frac{d^N}{d\tau^N} (\tau^2 - 1)^N. \quad (3)$$

The LGL points are defined as $\tau_0 = -1$, $\tau_N = 1$, and τ_k for being the roots of $P_N(\tau)$ for $k = 1, 2, \dots, N-1$. Clearly, these $(N+1)$ LGL points are also the roots of $(1-\tau^2)\dot{P}_N(\tau)$. Accurate calculation of LGL points is important for the successful implementation of pseudospectral transcription. Up to now, there is no explicit formula to compute the roots of

$\dot{P}_N(\tau)$. Instead, we adopt an iterative numerical method to compute these roots in section II-B.

The state variable $\mathbf{x}(\tau)$ and control variable $\mathbf{u}(\tau)$ are discretized at points $\{\tau_0, \tau_1, \dots, \tau_N\}$. The discretized state vectors are $\{\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_N\}$ and the discretized control vectors are $\{\mathbf{U}_0, \mathbf{U}_1, \dots, \mathbf{U}_N\}$, where $\mathbf{X}_i = \mathbf{x}(\tau_i)$ and $\mathbf{U}_i = \mathbf{u}(\tau_i)$. Then we approximate $\mathbf{x}(\tau)$ and $\mathbf{u}(\tau)$ using

$$\begin{aligned} \mathbf{x}(\tau) &\approx \mathbf{X}(\tau) = \sum_{i=0}^N L_i(\tau) \mathbf{X}_i, \\ \mathbf{u}(\tau) &\approx \mathbf{U}(\tau) = \sum_{i=0}^N L_i(\tau) \mathbf{U}_i, \end{aligned} \quad (4)$$

where $L_i(\tau)$ denotes Lagrange interpolating basis functions,

$$L_i(\tau) = \prod_{j=0, j \neq i}^N (\tau - \tau_j) / (\tau_i - \tau_j). \quad (5)$$

Step 3: State space equation transformation

The derivative of state variables is approximated by a differentiation matrix multiplied with discretized state variables, i.e.

$$\dot{\mathbf{x}}(\tau_k) \approx \dot{\mathbf{X}}(\tau_k) = \sum_{i=0}^N \dot{L}_i(\tau_k) \mathbf{X}_i = \sum_{i=0}^N D_{ki} \mathbf{X}_i, \quad (6)$$

where $k = 0, 1, 2, \dots, N$ and $\mathbf{D} = \{D_{ki}\} \in \mathbb{R}^{(N+1) \times (N+1)}$ denotes the differentiation matrix [11], defined as

$$D_{ki} = \begin{cases} \frac{P_N(\tau_k)}{P_N(\tau_i)(\tau_k - \tau_i)}, & i \neq k \\ -N(N+1)/4, & i = k = 0 \\ N(N+1)/4, & i = k = N \\ 0, & \text{otherwise} \end{cases}. \quad (7)$$

Now the state space equation can be enforced as the following $(N+1)$ equality constraints at LGL points

$$\sum_{i=0}^N D_{ki} \mathbf{X}_i - \frac{t_f - t_0}{2} \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, \tau_k) = 0. \quad (8)$$

Step 4: Performance index transformation

The performance index (1) is transformed using the Gauss-Lobatto quadrature rule

$$J = \phi(\mathbf{X}_N, \tau_N) + \frac{t_f - t_0}{2} \sum_{k=0}^N w_k G(\mathbf{X}_k, \mathbf{U}_k, \tau_k), \quad (9)$$

where w denotes the integration weight given by

$$w_k = \int_{-1}^1 l_k(\tau) d\tau = \frac{2}{N(N+1)P_N^2(\tau_k)}. \quad (10)$$

The Gauss-Lobatto quadrature is critical to ensure the accuracy of transforming the integral. It is well known that the residual in Gauss-Lobatto quadrature is [23]

$$R_{N+1} = \frac{-(N+1)N^3 2^{2N+1} ((N-1)!)^4}{(2N+1)(2N!)^3} G^{2N}(\xi). \quad (11)$$

Thus, by using $(N+1)$ LGL points, the quadrature residual is equal to zero for any polynomials with order less than $(2N-1)$.

Step 5: OCP to NLP problem conversion

Using aforementioned steps, the OCP is converted to the following NLP problem

$$\min_{\mathbf{X}_k, \mathbf{U}_k} J = \phi(\mathbf{X}_N, \tau_N) + \frac{t_f - t_0}{2} \sum_{k=0}^N w_k G(\mathbf{X}_k, \mathbf{U}_k, \tau_k). \quad (12)$$

s.t.

$$\begin{aligned} \left\| \sum_{i=0}^N D_{ki} \mathbf{X}_i - \frac{t_f - t_0}{2} \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, \tau_k) \right\|_{\infty} &\leq \delta, \\ \|\boldsymbol{\varphi}(\mathbf{X}_0, \mathbf{X}_N, \tau_N, \tau_0)\|_{\infty} &\leq \delta, \\ \|\mathbf{C}_{eq}(\mathbf{X}_k, \mathbf{U}_k, \tau_k)\|_{\infty} &\leq \delta, \\ \mathbf{C}_{ineq}(\mathbf{X}_k, \mathbf{U}_k, \tau_k) &\leq 0, \end{aligned}$$

where $k, i = 0, \dots, N$. The equality constraints are imposed at all points including two endpoints. A scalar δ is used to relax the equality constraints to avoid infeasibility. The scalar δ can be selected as [24]

$$\delta = (N - 1)^{\alpha - m}, \quad (13)$$

where m implies that the optimal state variable has continuous $(m - 1)$ th order classical derivatives, α is usually set as $3/2$. There are $(N_x + N_u) \times N$ optimization variables in NLP problem (12). For free terminal time t_f , we can also take t_f as an additional unknown variable. The converted NLP problem is a high-dimensional sparse optimization problem. Many known sparse NLP solvers can be used to solve this problem, e.g., SNOPT [25].

B. Numerical computation of collocation points

Since there is no explicit expression for the roots of $\dot{P}_N(\tau)$, directly solving $\dot{P}_N(\tau)$ through Legendre polynomials has very low accuracy. We employ the quasi-Newton algorithm to iteratively obtain the roots of $\dot{P}_N(\tau)$ [26]. The quasi-Newton method includes two main parts: choosing initial guess and designing the updating law. Let $g(z) = (1 - z^2)\dot{P}_N(z)$. The initial guess for iteration is chosen as the poles of N -order Chebyshev polynomial,

$$\mathbf{z}_0 = \{\theta_k | \theta_k = \cos(\pi k / N)\}, \quad (14)$$

where $k = 0, 1, \dots, N$. The updating law is designed as

$$\mathbf{z}_{n+1} = \mathbf{z}_n - \Delta \mathbf{z}_n, \quad (15)$$

where n denotes the step index, $\Delta \mathbf{z}_n$ is the length of each iteration. The $\Delta \mathbf{z}_n$ is mathematically expressed as

$$\Delta \mathbf{z}_n = (\mathbf{z}_n P_N(\mathbf{z}_n) - P_{N-1}(\mathbf{z}_n)) / (N P_N(\mathbf{z}_n)). \quad (16)$$

Note that (16) needs to compute the values of $P_N(\mathbf{z}_n)$ and $P_{N-1}(\mathbf{z}_n)$ by using the recursive relation of Legendre polynomial [23], i.e.

$$P_{m+1}(\mathbf{z}_n) = \frac{2m+1}{m+1} \mathbf{z}_n P_m(\mathbf{z}_n) - \frac{m}{m+1} P_{m-1}(\mathbf{z}_n), \quad (17)$$

where $m = 3, 4, \dots, N$. In summary, the computation algorithm for LGL collocation points is:

- 1) Initial points are determined using (14).
- 2) For step n ,
 - a) Recursively compute $P_N(\mathbf{z}_n)$ using (17) with initial conditions $P_1(\mathbf{z}_n) = 1; P_2(\mathbf{z}_n) = \mathbf{z}_n$;
 - b) Use (15) and (16) to iteratively compute \mathbf{z}_{n+1} .
- 3) Stop if $e = \|\mathbf{z}_{n+1} - \mathbf{z}_n\|_{\infty} < \varepsilon$.

The differentiation matrices and integration weights can be calculated by (7) and (10) respectively with accurate calculated LGL points.

III. PSEUDO-SPECTRAL OPTIMAL CONTROL PROBLEM SOLVER (POPS)

A. Introduction

A software package is developed in Matlab environment, called Pseudospectral Optimal control Problem Solver (POPS), to numerically calculate the optimal solution of generic OCPs. It consists of four modules: 1) User configuration module including OCP description and solver parameter setup; 2) Module for converting OCP into NLP problem; 3) Module to solve NLP problem; 4) Module for interacting with users, including data saving, figure drawing, etc. The POPS is applicable to solve OCPs with: 1) both linear and nonlinear; 2) smooth functions or non-smoothing functions that can be written in multi-phase form; 3) free, fixed, or constrained initial state, terminal state and terminal time; 4) equality or inequality constraints, differentiation-typed or integration-typed constraints; 5) Mayer, Lagrange, or Bolza-typed performance index. The POPS is compatible with other Matlab toolboxes, and users can call POPS in other Matlab applications. It also provides a large amount of flexibility for users to configure their own problems. The authors can provide a free copy of POPS to interested readers for non-commercial purposes. If necessary, one can send the requests to the corresponding author (Dr. Shengbo Eben Li, lisb04@gmail.com) for a free copy.

B. Example for POPS

To verify the performance of POPS, a high order nonlinear OCP about trajectory optimization for aircraft is selected. The study of the numerical method for solving this kind of problem started as early as 1964 [27]. The optimization problem of this case study is formulated as

$$\min J = -r(t_f). \quad (18)$$

s.t.

$$\begin{aligned} \dot{r} &= v_r, \\ \dot{\theta} &= v_{\theta}/r, \\ \dot{v}_r &= v_{\theta}^2/r - \mu/r^2 + a \sin(\beta), \\ \dot{v}_{\theta} &= -v_r v_{\theta}/r + a \cos(\beta), \\ [r, \theta, v_r, v_{\theta}]_{t_0=0} &= [1 \ 0 \ 0 \ 1], \\ [v_r, v_{\theta}]_{t_f} &= [0, (\mu/r_{t_f})^{0.5}], \\ a &= T/(m_0 - |\dot{m}|t), \end{aligned}$$

where $\mu = 1$, $T = 0.1405$, $m_0 = 1$, $|\dot{m}| = 0.0749$, and $t_f = 3.32$. This problem is strongly nonlinear and also time-varying. It is difficult to obtain its optimal solution by constructing and solving differential equations through the first-order necessary condition.

The POPS is employed below to solve this problem. We first verify the efficiency for computing LGL collocation points as described in section II-B. For the case study considered here, we run POPS in Matlab 2012a with 3.2 GHz CPU, and set $\varepsilon = 10^{-1}$. Computation time for choosing 40, 70, or 500 collocation points is 22 ms, 31 ms, or 35 ms, respectively. This means the computational efficiency for obtaining LGL collocation points based on quasi-Newton method satisfies the common requirements of computational resources.

In the following we choose 70 collocation points for running POPS to solve the optimization problem (18). The optimal trajectories of state and control variables obtained from POPS are shown in Fig. 1. The total CPU time for running POPS is 2.54 seconds. The optimal performance index J is obtained as -1.52527, which is consistent with the value -1.525 obtained using DIDO solver [28]. In Fig. 2, we depict the performance indexes obtained from POPS with different number of collocation points. It is observed from Fig. 2 that the performance index nearly approaches the optimal value as the number of collocation points is greater than 40, which is also consistent with the observations from [28].

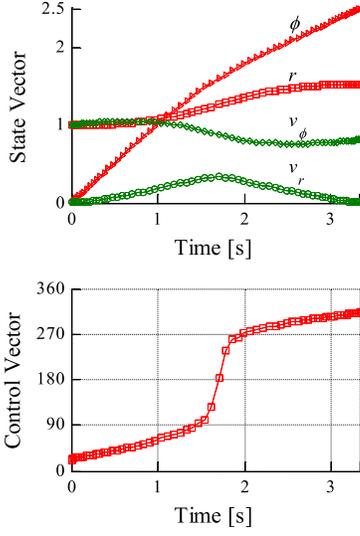


Figure 1. Optimal trajectories for OCP (18)

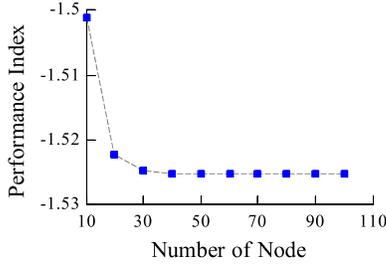


Figure 2. Optimal J with different numbers of nodes

IV. APPLICATION TO FUEL-OPTIMIZED AUTOMATED VEHICLES

For ground vehicles, fuel-saving techniques have been enthusiastically studied over the past decades. One important topic is the eco-driving strategy. It was demonstrated that driving styles can change fuel efficiency by up to 10% in normal traffic [29]. The fuel-optimized automation aims to implement proper control to vehicles to minimize fuel consumption during travelling. The design of such strategies is in essential an OCP. This section considers a typical case study, i.e. cruise control in hilly road conditions. The used vehicle is a passenger car equipped with a 2.0L gasoline engine and a continuous variable transmission (CVT).

A. Fuel-optimized cruise control on hilly road

The vehicle runs on a hilly road with both uphill and downhill. As shown in Fig. 3, road AB is flat with length L_1 , road BC is a hilly section with horizontal length L_2 and height H , and road CD is flat with length L_3 . There is a traffic light at the end point D such that the green light is only on for $t \in [t_{min}, t_{max}]$. The goal is to travel from point A to point D with minimal fuel consumption while subjecting to the constraints that vehicle must pass through point D during $t_f \in [t_{min}, t_{max}]$ and the velocity should be bounded in $v \in [v_{min}, v_{max}]$.

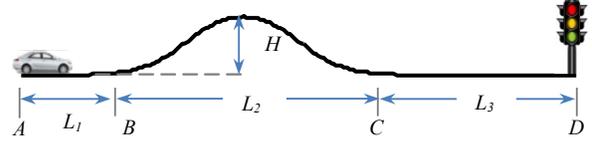


Figure 3. Setup of fuel optimized cruise control problem

For the passenger car, the longitudinal dynamic model is given as

$$i_g i_0 \eta_T T_e / r_w = \delta M \dot{v} + F_R, \quad (19)$$

$$F_R = 0.5 C_D \rho_a A_v v^2 + M g (f \cos \theta_s + \sin \theta_s),$$

where i_g denotes the transmission ratio, i_0 denotes the final gear ratio, η_T denotes the driveline efficiency, T_e denotes the engine torque, δ denotes rotating mass coefficient, M denotes the total mass of vehicle, C_D denotes aerodynamic drag coefficient, ρ_a denotes air density, A_v denotes frontal area of vehicle, v denotes vehicle velocity, f denotes the rolling resistance coefficient, θ_s denotes the road slope. By assuming no clutch sliding, the following relationship between engine speed w_e and vehicle velocity v holds

$$w_e = 60 v i_g i_0 / (2\pi r). \quad (20)$$

The distance s , velocity v , and acceleration a are related as

$$\begin{aligned} \dot{s} &= v \cos \theta_s, \\ \dot{v} &= a. \end{aligned} \quad (21)$$

The engine fuel injection rate is an analytical model coming from the least-square fitting of engine brake specific fuel consumption (BSFC) map

$$F_s(T_e, w_e) = \sum_{i=0}^4 \sum_{j=0}^i A_{1+j+\sum_0^i} \left(\frac{T_e}{200}\right)^{i-j} \left(\frac{w_e}{650}\right)^j, \quad (22)$$

where A is the fitting coefficients. The engine BSFC curve is shown as Fig. 4. The optimal BSFC line (i.e., E-line) of engine is fitted as

$$T_{eco} = k_{eco} (w_{eco} - 1000)^\gamma, \quad (23)$$

where k_{eco} is the fitting coefficient, and $\gamma = 1/3$ is the exponent of fitting function. The optimal E-line is also shown in Fig. 4. It is assumed that the CVT ratio is accurately controlled such that the engine always works along the optimal BSFC line for any output power.

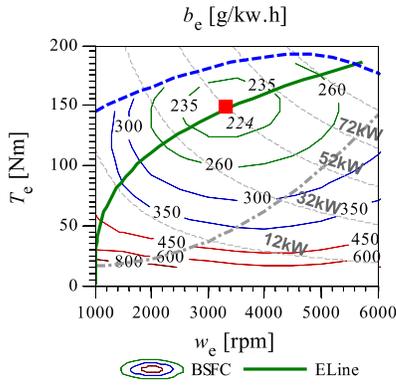


Figure 4. Brake specific fuel consumption map

The OCP is formulated as

$$\min J = \int_0^{t_f} F_s + k_e (dT_e/dt)^2 dt. \quad (24)$$

s.t.

$$\begin{aligned} \dot{s} &= v \cos \theta_s, \\ \dot{v} &= i_g i_0 \eta_T T_e / (\delta M r) - F_R / \delta M, \\ w_e &= 60 v i_g i_0 / (2 \pi r), \\ T_e - k_{eco} (w_e - 1000)^\gamma &= 0, \\ w_{e\min} &\leq w_e \leq w_{e\max}, \\ i_{g\min} &< i_g \leq i_{g\max}, \\ v_{\min} &\leq v \leq v_{\max}, \\ t_{\min} &\leq t_f \leq t_{\max}, \\ s(0) &= 0, \\ s(t_f) &= L_1 + L_2 + L_3, \\ v(0) &= v_0, \end{aligned}$$

where state variables include the distance s and velocity v , and control variables include engine torque T_e and transmission ratio i_g . The parameters are listed in TABLE. I.

TABLE. I SIMULATION PARAMETERS

Para	Value	Para	Value
C_D	0.316	L_1	50 m
A_v	2.22 m ²	L_2	200 m
ρ_a	1.226 kg/m ³	L_3	100 m
M	1600 kg	H	8 m
f	0.028	v_0	25/50/70 km/h
δ	1.2	v_{\min}	15 km/h
i_0	3.863	v_{\max}	80 km/h
η_T	0.9	t_{\min}	20 s
r	0.307 m	t_{\max}	35 s
$w_{e\min/\max}$	1000/6000 rpm	k_e	0.005
$i_{g\min/\max}$	0.4/2.6		

The POPS is applied to solve the OCP (24) under the condition of 50 collocation points. Three different initial velocities are considered, 25 km/h, 50km/h, and 75km/h. The optimization results are given below. For three different initial velocities, the fuel consumptions are 37.71g, 27.22g, and 16.75g, respectively; the travelling time to point D is 33.65s, 30.17s, and 23.92s, respectively. Fig. 5 depicts the solutions of distance s , velocity v , and engine torque T_e .

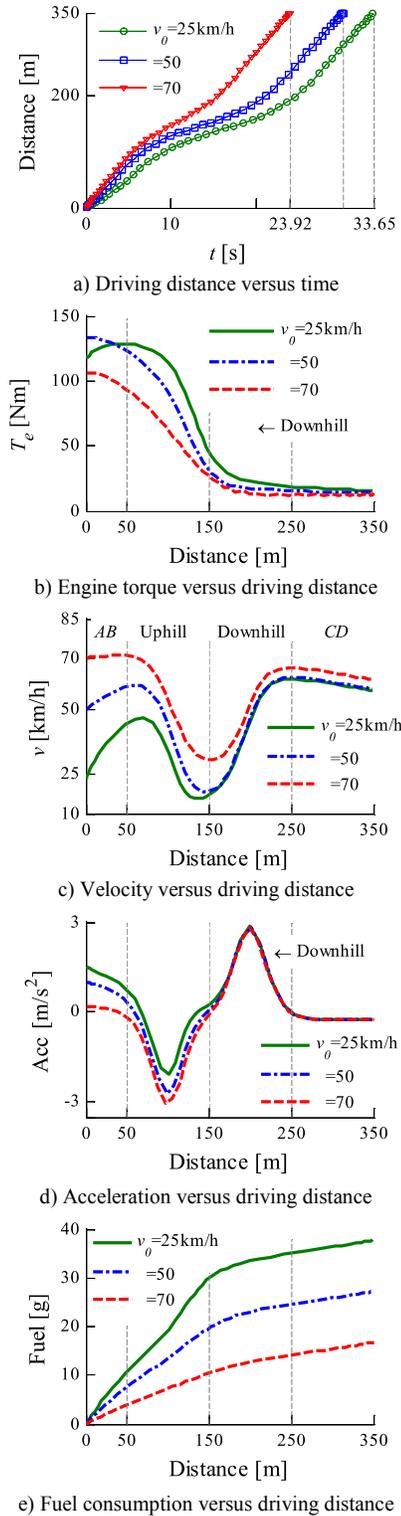


Figure 5. POPS optimization results

It is observed from Fig. 5 that on road AB, the vehicle keeps accelerating to a high speed; afterwards, engine gradually reduces its power to slow down vehicle. On the uphill, the vehicle velocity drops quickly; on the downhill, the vehicle regains high speed; on road CD, the engine torque remains small to let the vehicle coast down to point D. During driving on uphill, kinetic energy is transformed into potential

energy, thus largely reducing the vehicle velocity. During driving on downhill, the vehicle releases potential energy to accelerate.

V. CONCLUSION

In this paper, we proposed the computational framework and implementation of Legendre PS method and applied it to obtain the fuel-optimized strategies. The PS method has lower sensitivity to initial guess, faster convergence rate, and higher accuracy compared to traditional direct methods, which has the potential to promote the development of vehicle optimal control, e.g. power management for HEVs/EVs, optimal control for active suspensions, trajectory optimization for automated vehicles, and fuel optimized eco-driving. The main idea of PS method is to convert the OCP into a NLP problem, which is then numerically solved by matured optimization algorithms. A software package (called POPS) is developed using Legendre PS method in Matlab environment, suitable for various types of OCPs. It is compatible with other Matlab toolboxes and free to users with non-commercial purposes. An example of obtain optimal strategies for automated vehicles is used to verify the performance of POPS.

REFERENCES

- [1] S. Delprat, J. Lauber, T. M. Guerra, and J. Rimaux, "Control of a parallel hybrid powertrain: optimal control," *IEEE Trans. Vehicular Technology*, vol. 53, no. 3, pp. 872–881, 2004.
- [2] S. E. Li, H. Peng, K. Li, and J. Wang, "Minimum Fuel Control Strategy in Automated Car-Following Scenarios," *IEEE Trans. Vehicular Technology*, vol. 61, no. 3, pp. 998–1007, 2012.
- [3] M. Kuriyama, S. Yamamoto, and M. Miyatake, "Theoretical study on eco-driving technique for an electric vehicle with dynamic programming," in *2010 IEEE Int. Conf. Electrical Machines and Systems (ICEMS)*, pp. 2026–2030.
- [4] M. A. S. Kamal, M. Mukai, J. Murata, and T. Kawabe, "Ecological vehicle control on roads with up-down slopes," *IEEE Trans. Intelligent Transportation Systems*, vol. 12, no. 3, pp. 783–794, 2011.
- [5] F. Mensing, R. Trigui, and E. Bideaux, "Vehicle trajectory optimization for application in ECO-driving," in *Vehicle Power and Propulsion Conference (VPPC), IEEE*, 2011, pp. 1–6.
- [6] L. Nouveliere, M. Braci, L. Menhour, et al., "Fuel consumption optimization for a city bus," in *UKACC Control Conference*, 2008, pp. 1–6.
- [7] D. G. Hull, "Conversion of optimal control problems into parameter optimization problems," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 1, pp. 57–60, 1997.
- [8] D. Blackwell, "Discounted dynamic programming," *The Annals of Mathematical Statistics*, pp. 226–235, 1965.
- [9] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*. PA USA: SIAM, 2010, pp. 91–118.
- [10] G. Elnagar, M. A. Kazemi, and M. Razzaghi, "The pseudospectral Legendre method for discretizing optimal control problems," *IEEE Trans. Automatic Control*, vol. 40, no. 10, pp. 1793–1796, 1995.
- [11] F. Fahroo and I. M. Ross, "Advances in pseudospectral methods for optimal control," in *Proc. AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, Hawaii, August 2008, pp. 18–21.
- [12] F. Fahroo and I. M. Ross, "Costate estimation by a Legendre pseudospectral method," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 2, pp. 270–277, 2001.
- [13] Q. Gong, W. Kang, N. S. Bedrossian, et al., "Pseudospectral optimal control for military and industrial applications," in *Proc. 46th IEEE Conf. Decision and Control*, December 2007, pp. 4128–4142.
- [14] F. Fahroo and I. M. Ross, "Pseudospectral methods for infinite-horizon nonlinear optimal control problems," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 4, pp. 927–936, 2008.
- [15] D. Garg, W. W. Hager, and A. V. Rao, "Pseudospectral methods for solving infinite-horizon optimal control problems," *Automatica*, vol. 47, no. 4, pp. 829–837, 2011.
- [16] I. M. Ross and F. Fahroo, "Convergence of pseudospectral discretizations of optimal control problems," in *Proc. 40th IEEE Conf. Decision and Control*, vol. 4, 2001, pp. 3175–3177.
- [17] L. N. Trefethen, *Spectral methods in MATLAB*. PA USA: SIAM, 2000, pp. 7–40.
- [18] G. T. Hunting, D. Benson, and A. V. Rao, "A comparison of accuracy and computational efficiency of three pseudospectral methods," in *Proc. the AIAA Guidance, Navigation, and Control Conference*, Hilton Head, SC, 2007.
- [19] I. M. Ross, "User's manual for DIDO: A MATLAB application package for solving optimal control problems," *Tomlab Optimization, Sweden*, 2004.
- [20] A. V. Rao, D. A. Benson, C. Darby, et al., "Algorithm 902: A matlab software for solving multiple-phase optimal control problems using the gauss pseudospectral method," *ACM Transactions on Mathematical Software*, vol. 37, no. 2, 2010.
- [21] V. M. Becerra, "PSOPT optimal control solver user manual," University of Reading, 2010.
- [22] P. E. Rutquist and M. M. Edvall, "Propt-matlab optimal control software," *Tomlab Optimization Inc 260*, 2009.
- [23] P. G. Ciarlet and J. L. Lions, *Handbook of numerical analysis* (Vol. 8). Gulf Professional Publishing, 2002.
- [24] Q. Gong, I. M. Ross, W. Kang, and F. Fahroo, "Connections between the covector mapping theorem and convergence of pseudospectral methods for optimal control," *Computational Optimization and Applications*, vol. 41, no. 3, pp. 307–335, 2008.
- [25] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM journal on optimization*, vol. 12, no. 4, pp. 979–1006, 2002.
- [26] Dennis, J. E. Jr, and J. J. Moré, "Quasi-Newton methods, motivation and theory," *Siam review*, vol. 19, no. 1, pp. 46–89, 1977.
- [27] H. G. Moyer and G. Pinkham, "Several Trajectory Optimization Techniques. Part II: Application," *Grumman Aircraft Engineering Corp Bethpage NY*, 1963.
- [28] J. Strizzi, I. M. Ross, and F. Fahroo, "Towards real-time computation of optimal controls for nonlinear systems," in *Proc. the AIAA Guidance, Navigation, and Control Conference*, Monterey, CA, 2002, pp. 5–8.
- [29] J. N. Barkenbus, "Eco-driving: An overlooked climate change initiative," *Energy Policy*, vol. 38, no. 2, pp. 762–769, 2010.