

A Unified Pseudospectral Computational Framework for Optimal Control of Road Vehicles

Shaobing Xu, Shengbo Eben Li, *Member, IEEE*, Kun Deng, *Member, IEEE*, Sisi Li, *Student Member, IEEE*, and Bo Cheng

Abstract—This paper presents a unified pseudospectral computational framework for accurately and efficiently solving optimal control problems (OCPs) of road vehicles. Under this framework, any continuous-time OCP is converted into a nonlinear programming (NLP) problem via pseudospectral transformation, in which both states and controls are approximated by global Lagrange interpolating polynomials at Legendre–Gauss–Lobatto (LGL) collocation points. The mapping relationship between the costates of OCP and the KKT multipliers of NLP is derived for checking the optimality of solutions. For the sake of engineering practice, a quasi-Newton iterative algorithm is integrated to accurately calculate the LGL points, and a multiphase preprocessing strategy is proposed to handle nonsmooth problems. A general solver called pseudospectral OCP solver (POPS) is developed in MATLAB environment to implement the computational framework. Finally, two classic vehicle automation problems are formulated and numerically solved by POPS: 1) optimization of ecodriving strategy in hilly road conditions; and 2) optimal path planning in an overtaking scenario. The comparison with an equally spaced direct method is presented to show the effectiveness of this unified framework.

Index Terms—Ecodriving, intelligent vehicle, optimal control, path planning, pseudospectral (PS) method.

I. INTRODUCTION

As customers become more demanding on comfort, safety, and efficiency, road vehicles gain more popularity in terms of automation, electrification, and hybridization [1], [2]. One common question is how to design optimal strategies to maximally enhance some performances of road vehicles. This question becomes more challenging due to the nonlinearity and uncertainty of vehicle dynamics, limited computing resources of control units, time-varying road/traffic conditions, etc. In automotive engineering, typical optimal control examples include power management for hybrid electric vehicles (HEVs)/electric

vehicles [3], [4], optimal control for active suspensions [5]–[7], trajectory optimization for automated vehicles [8], fuel-optimized ecodriving assistance [9], etc. The power management for HEV is referred to dynamically optimizing the split of power between thermal and electrical paths, in pursuit of energy-saving for vehicle driving [1]. Active suspensions are in the sense to achieve optimal performance of ride and handling via dynamically manipulating stiffness and damping of suspensions in response to vehicle motions and road conditions [5]. The trajectory planning for automated vehicles concerns the real-time optimization of vehicle routes to accomplish the designated driving tasks [8]. The fuel-optimized ecodriving assistance aims to reduce fuel consumption by dynamically adjusting throttle angle, transmission gear ratio, and brake pressure in accordance with constraints on traffic conditions, road conditions, and the ability of vehicles [9].

In essence, obtaining the aforementioned optimal strategies is associated with optimal control problems (OCPs). Namely, it aims to minimize the given performance index of a dynamic system by manipulating control inputs. The main methods to obtain optimal strategies for both ICE-based and e-Powertrain-based road vehicles include Pontryagin’s minimum principle, dynamic programming, and traditional direct method (TDM) [1], [5], [10]–[13]. Pontryagin’s minimum principle falls into the category of indirect methods, and it gives the first-order necessary condition in the form of boundary value problems. When dealing with nonlinearities and complex constraints, this method is often incapable of obtaining optimal solutions in an efficient manner [14]. Dynamic programming is widely applied to OCPs that are difficult to be solved analytically; however, this method suffers from the curse of dimensionality and has low computational efficiency [12], [13], [15]. The TDMs, such as shooting methods and collocation methods, are also widely used for handling complex problems in automation, hybridization, and electrification of road vehicles. Compared to the TDMs, the pseudospectral (PS) method is more attractive due to its merits on high accuracy, lower sensitivity to initial value and faster convergence [16].

In 1995, Elnagar *et al.* first introduced the concept of the PS method into optimal control community [17]. Since 2000, Fahroo, Ross, Gong, Rao *et al.* intensively studied the PS method, and obtained a series of important achievements, e.g., the existence of optimal solution, covector mapping theorem, convergence rate theorem, and principles for choosing the consistency parameter and the interpolation weight function [16], [18]–[21]. The main idea of the PS method is to approximate state and control variables through a finite order of global

Manuscript received May 14, 2014; revised August 24, 2014; accepted September 17, 2014. Date of publication October 16, 2014; date of current version August 12, 2015. Recommended by Guest Editor H. Gao. This work was supported by National Science Foundation China under Grant 51205228 and Tsinghua University Initiative Scientific Research Program under Grant 2012THZ0.

S. Xu, S. E. Li, and B. Cheng are with the State Key Laboratory of Automotive Safety and Energy, Department of Automotive Engineering, Tsinghua University, Beijing 100084, China (e-mail: xsbing2008@foxmail.com; lisb04@gmail.com; chengbo@tsinghua.edu.cn).

K. Deng is with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: kundeng2@gmail.com).

S. Li is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: sophieandli@gmail.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TMECH.2014.2360613

interpolating polynomials and convert the OCP into a nonlinear programming (NLP) problem at orthogonal collocation points [16], [17]. The formulated NLP problem has been shown to converge to the original OCP in spectral accuracy [22]. For any infinitely differentiable function, the spectral accuracy is $O(N^{-m})$ for every $m \in \mathbb{Z}^+$, where N denotes N th-order approximation. For any analytic function, the convergence will be faster at the rate of (c^N) for some constant $0 < c < 1$ [23]. Nowadays, the PS method has been successfully applied to problems in aerospace engineering. Some PS-based OCP solvers are also commercially available for researchers, e.g., DIDO by Ross *et al.* [24], GPOPS by Rao *et al.* [25], PSOPT by Becerra *et al.* [26], and PROPT by Tomlab Optimization [27].

There are several typical PS methods: Chebyshev pseudospectral method (CPM), Legendre pseudospectral method (LPM), Gauss pseudospectral method (GPM), and Radau pseudospectral method (RPM). The CPM adopts Chebyshev–Gauss–Lobatto points to discretize states/inputs and uses Clenshaw–Curtis quadrature for numerical integration [28]. The LPM, GPM, and RPM use Lagrange interpolating polynomials to approximate states and control variables and Gaussian integral to calculate cost functions. The major difference among the three methods is the selection of collocation points, but all three kinds of points are calculated based on Legendre polynomial [17], [29], [30]. The comparison of the performance of LPM, RPM, and GPM indicates that Gauss and Radau methods have similar computational accuracy, and they have better costate estimation capabilities than the Legendre method [32]. However, the Legendre method has better performance for OCPs with fixed boundary conditions, while Gauss and Radau methods may not converge [16], [31].

The main contribution of this paper is to develop a unified computational framework and software package aiming to efficiently calculate different types of OCPs arising from automotive engineering. The computational framework is based on the LPM and three new improvements, i.e., 1) performing the costate estimation for a more general OCP with a Bolza-type performance index and both equality and inequality constraints; 2) adopting a quasi-Newton iterative algorithm to efficiently calculate collocation points; and 3) proposing a method to segment and transform nonsmooth problems. In Section II, we introduce the framework to solve OCPs. In Section III, we discuss some important implementation issues of LPM, including costate estimation, accurate calculation of Legendre–Gauss–Lobatto (LGL) points, multiphase preprocessing for nonsmooth problems; a MATLAB environment-based software package is also developed. In Section IV, we apply the developed software package to solve two typical vehicle automation problems, i.e., ecodriving problem and path planning problem. The performance of this unified framework is finally compared with a traditional equally spaced direct method.

II. COMPUTATIONAL FRAMEWORK OF THE LPM

In automotive engineering, many problems aim to find an optimal control law $\mathbf{u}(t) \in \mathbb{R}^{N_u}$ to minimize a designed performance index. For example, in economical automation systems,

we usually take the engine torque or acceleration as the control inputs and expect finding an optimal solution to minimize fuel consumption for a specific driving task. Other similar problems include the charge/discharge management of batteries, energy distributions for HEVs, optimizations of the stiffness and resistance for active suspensions, and path planning for automated vehicles. All these practical problems can be expressed by a general Bolza-type OCP with terminal constraints, equality and inequality path constraints, i.e.

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

subject to

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

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 space function, $\emptyset(\cdot)$ denotes the Mayer performance index, $G(\cdot)$ denotes the Lagrange performance index, $\varphi(\cdot)$ denotes the initial and terminal constraints, $\mathbf{C}_{\text{eq}}(\cdot)$ and $\mathbf{C}_{\text{inq}}(\cdot)$ denote the equality and inequality path constraints, respectively.

Using the PS transformation, we approximate both state and control variables by Lagrange interpolating polynomials at the LGL points. The state-space equations are represented as equality constraints. The integral of cost function is calculated by the Gauss–Lobatto quadrature rule. Then, the OCP is converted into a NLP problem, which can be solved by available optimization solvers. The detailed framework is described next.

Step 1: Time-domain transformation. For the sake of simplicity, we first transform the time domain $[t_0, t_f]$ to the canonical interval $[-1, 1]$:

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

Step 2: Collocation and discretization. For PS methods, the collocation points often come from the roots of orthogonal polynomials, which help avoid the Runge phenomenon. Let $P_N(\tau)$ denote the N -order Legendre polynomials [17], 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 being the roots of $\dot{P}_N(\tau)$ 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 to successfully implement the PS transformation. However, there is no explicit formula to compute the roots of $\dot{P}_N(\tau)$. Instead, we introduce an iterative numerical algorithm to compute these roots in Section III-B.

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

$\mathbf{U}_i = \mathbf{u}(\tau_i)$. Then, we approximate $\mathbf{x}(\tau)$ and $\mathbf{u}(\tau)$ by

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

where $L_i(\tau)$ denotes the Lagrange interpolating basis function

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

Step 3: Transformation of state space equation. The states are essentially approximated by interpolating polynomials. Then, the differential operation of states can be approximated by the differential operation on Lagrange bases, denoted as

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

where $k = 0, 1, 2, \dots, N$ and $\mathbf{D} = \{D_{ki}\} \in \mathbb{R}^{(N+1) \times (N+1)}$ denotes the differentiation matrix [16], defined as

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

Then, the state space equation can be converted 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: Transformation of performance index. The performance index is transformed using the Gauss–Lobatto quadrature rule

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

where w denotes the integration weight, defined as

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

The Gauss–Lobatto quadrature is critical to the accuracy of transforming the integral. The residual in Gauss–Lobatto quadrature is [33]

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

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, we can convert the OCP to the following NLP problem, i.e.

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

subject to

$$\begin{aligned}\left\| \sum_{i=0}^N D_{ki} \mathbf{X}_i - \frac{t_f - t_0}{2} \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, \tau_k) \right\|_{\infty} &\leq \delta \\ \|\varphi(\mathbf{X}_0, \mathbf{X}_N, \tau_0, \tau_N)\|_{\infty} &\leq \delta \\ \|\mathbf{C}_{\text{eq}}(\mathbf{X}_k, \mathbf{U}_k, \tau_k)\|_{\infty} &\leq \delta \\ \mathbf{C}_{\text{inq}}(\mathbf{X}_k, \mathbf{U}_k, \tau_k) &\leq 0,\end{aligned}\quad (12)$$

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 [19]

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

where m implies that the optimal state variable has continuous $(m-1)$ th-order classical derivatives, α is usually set as $3/2$ [19]. There are $(N_x + N_u) \times N$ variables to be optimized in the NLP problem (12). For nonfixed terminal time t_f , we can also take t_f as an additional variable to be optimized. The converted NLP is a high-dimensional and sparse problem. Some known sparse NLP solvers can be used to solve this kind of problem, like SNOPT used here [34].

III. IMPLEMENTATION OF THE PS METHOD

In this section, some important topics on implementing the method are discussed. We first present the costate estimation for the defined Bolza-type OCP with both equality and inequality constraints. Since the computation of LGL points has no explicit expression, a quasi-Newton iterative algorithm is introduced to numerically calculate the collocation points. A multiphase pre-processing strategy is proposed to deal with non-smooth problems, which is often accompanied with low accuracy if using the PS method directly. A MATLAB environment-based solver, i.e., pseudospectral optimal control problem solver (POPS), is developed finally.

A. Costate Estimation

The costate of OCPs plays an important role in evaluating the optimality of solutions, but direct methods cannot provide explicit expressions of costates. Ross *et al.* have suggested that there exists a mapping relationship between costate variables of the original OCP and KKT multipliers of the associated NLP [18]. This mapping relationship can be used to compute the costate variables indirectly from the KKT multipliers. In the following, we will derive the mapping relationship for Bolza-type OCP as shown in (1).

First, we construct the Hamiltonian for the OCP:

$$H = \frac{t_f - t_0}{2} (G + \boldsymbol{\lambda}^T \mathbf{f}) + \boldsymbol{\mu}_{\text{eq}}^T \mathbf{C}_{\text{eq}} + \boldsymbol{\mu}_{\text{inq}}^T \mathbf{C}_{\text{inq}}, \quad (14)$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{N_x}$, $\boldsymbol{\mu}_{\text{eq}} \in \mathbb{R}^{N_{\text{Ceq}}}$, $\boldsymbol{\mu}_{\text{inq}} \in \mathbb{R}^{N_{\text{Cinq}}}$ are costate variables of the OCP corresponding to the state space model, equality path constraints, and inequality path constraints. According to the first-order optimality condition, we have the following

differential equation:

$$\dot{\lambda}(\tau_k) = - \left(\frac{\partial H}{\partial x} \right) (\tau_k). \quad (15)$$

Since $\dot{\lambda}$ satisfies $\dot{\lambda}(\tau_k) = \sum_{i=0}^N D_{ki} \lambda(\tau_i)$, then

$$\begin{aligned} \frac{t_f - t_0}{2} \left[\frac{\partial G}{\partial x} + \left(\frac{\partial f}{\partial x} \right)^T \lambda(\tau_k) \right] + \left(\frac{\partial C_{eq}}{\partial x} \right)^T \mu_{eq} \\ + \left(\frac{\partial C_{inq}}{\partial x} \right)^T \mu_{inq} = - \sum_{i=0}^N D_{ki} \lambda(\tau_i). \end{aligned} \quad (16)$$

After applying the PS transformation, the Lagrangian function of associated NLP is given by

$$\begin{aligned} \tilde{J} = \emptyset + \frac{t_f - t_0}{2} \sum_{i=0}^N w_i G_i + \tilde{v}^T \varphi + \sum_{i=0}^N \left[\tilde{\lambda}_i^T \left(\frac{t_f - t_0}{2} f_i - \dot{X}_i \right) \right. \\ \left. + \tilde{\mu}_{eq,i}^T C_{eq,i} + \tilde{\mu}_{inq,i}^T C_{inq,i} \right], \end{aligned} \quad (17)$$

where $\tilde{\lambda}$, $\tilde{\mu}_{eq}$, $\tilde{\mu}_{inq}$, \tilde{v} are the KKT multipliers of the NLP. According to the KKT conditions for a generic NLP, we have

$$\begin{aligned} \frac{\partial \tilde{J}}{\partial X_k} = 0, \quad \frac{\partial \tilde{J}}{\partial U_k} = 0 \\ C_{eq,k} = 0, \quad \tilde{\mu}_{inq,k}^T C_{inq,k} = 0. \end{aligned} \quad (18)$$

The partial derivatives of \tilde{J} with respect to X_k are given next for $k = 1, \dots, N-1$:

$$\begin{aligned} \frac{\partial \tilde{J}}{\partial X_k} = \frac{t_f - t_0}{2} \left[\frac{\partial G_k}{\partial X_k} w_k + \left(\frac{\partial f_k}{\partial X_k} \right)^T \tilde{\lambda}_k \right] \\ + \left(\frac{\partial C_{eq,k}}{\partial X_k} \right)^T \tilde{\mu}_{eq,k} + \left(\frac{\partial C_{inq,k}}{\partial X_k} \right)^T \tilde{\mu}_{inq,k} \\ - \frac{\partial}{\partial X_k} \sum_{i=0}^N \tilde{\lambda}_i^T \dot{X}_i = 0 \end{aligned} \quad (19)$$

where

$$\begin{aligned} \frac{\partial}{\partial X_k} \sum_{i=0}^N \tilde{\lambda}_i^T \dot{X}_i = \sum_{i=0}^N \tilde{\lambda}_i^T \left(\frac{\partial}{\partial X_k} \sum_{n=0}^N D_{in} X_n \right) \\ = \sum_{i=0}^N D_{ik} \tilde{\lambda}_i. \end{aligned} \quad (20)$$

On the other hand, by considering

$$\begin{aligned} w_i D_{ik} = -w_k D_{ki}, \quad i \neq k \\ D_{ik} = D_{ki} = 0, \quad i = k. \end{aligned} \quad (21)$$

We have

$$\begin{aligned} \frac{t_f - t_0}{2} \left[\frac{\partial G_k}{\partial X_k} + \left(\frac{\partial f_k}{\partial X_k} \right)^T \frac{\tilde{\lambda}_k}{w_k} \right] + \left(\frac{\partial C_{eq,k}}{\partial X_k} \right)^T \frac{\tilde{\mu}_{eq,k}}{w_k} \\ + \left(\frac{\partial C_{inq,k}}{\partial X_k} \right)^T \frac{\tilde{\mu}_{inq,k}}{w_k} = - \sum_{i=0}^N D_{ki} \frac{\tilde{\lambda}_i}{w_i}. \end{aligned} \quad (22)$$

By comparing (22) with (16), we know that if two equations are equivalent, we must have

$$\lambda(\tau_k) = \frac{\tilde{\lambda}_k}{w_k}, \quad k = 1, \dots, N-1 \quad (23)$$

Equation (23) provides the mapping relations between $\lambda(\tau_k)$ and $\tilde{\lambda}_k$ for $k = 1, \dots, N-1$. For $k = 0, N$, we need to consider the effects of Mayer function \emptyset and constraint φ to derive mapping relations for $\lambda(t_0)$ and $\lambda(t_N)$. The partial derivative of \tilde{J} with respect to X_0 is given by

$$\begin{aligned} \frac{t_f - t_0}{2} \left[\frac{\partial G_0}{\partial X_0} w_0 + \left(\frac{\partial f_0}{\partial X_0} \right)^T \tilde{\lambda}_0 \right] + \left(\frac{\partial C_{eq,0}}{\partial X_0} \right)^T \tilde{\mu}_{eq,0} \\ + \left(\frac{\partial C_{inq,0}}{\partial X_0} \right)^T \tilde{\mu}_{inq,0} + \left(\frac{\partial \varphi}{\partial X_0} \right)^T \tilde{v} - \sum_{i=0}^N D_{i0} \tilde{\lambda}_i = 0. \end{aligned} \quad (24)$$

Since

$$\begin{aligned} w_i D_{i0} = -w_0 D_{0i} \quad i \neq 0 \\ D_{00} = -1/(2w_0) \quad i = 0. \end{aligned} \quad (25)$$

Then

$$\sum_{i=0}^N D_{i0} \tilde{\lambda}_i = -w_0 \sum_{i=0}^N D_{0i} \frac{\tilde{\lambda}_i}{w_i} - \frac{\tilde{\lambda}_0}{w_0} \quad (26)$$

Thus, we have

$$\begin{aligned} \frac{t_f - t_0}{2} \left[\frac{\partial G_0}{\partial X_0} + \left(\frac{\partial f_0}{\partial X_0} \right)^T \frac{\tilde{\lambda}_0}{w_0} \right] + \left(\frac{\partial C_{eq,0}}{\partial X_0} \right)^T \frac{\tilde{\mu}_{eq,0}}{w_0} \\ + \left(\frac{\partial C_{inq,0}}{\partial X_0} \right)^T \frac{\tilde{\mu}_{inq,0}}{w_0} = - \sum_{i=0}^N D_{0i} \frac{\tilde{\lambda}_i}{w_i} \\ - \frac{1}{w_0} \left[\frac{\tilde{\lambda}_0}{w_0} + \left(\frac{\partial \varphi}{\partial X_0} \right)^T \tilde{v} \right]. \end{aligned} \quad (27)$$

Similar arguments hold when considering the partial derivative of \tilde{J} with respect to X_N :

$$\begin{aligned} \frac{t_f - t_0}{2} \left[\frac{\partial G_N}{\partial X_N} + \left(\frac{\partial f_N}{\partial X_N} \right)^T \frac{\tilde{\lambda}_N}{w_N} \right] + \left(\frac{\partial C_{eq,N}}{\partial X_N} \right)^T \frac{\tilde{\mu}_{eq,N}}{w_N} \\ + \left(\frac{\partial C_{inq,N}}{\partial X_N} \right)^T \frac{\tilde{\mu}_{inq,N}}{w_N} = - \sum_{i=0}^N D_{Ni} \frac{\tilde{\lambda}_i}{w_i} \\ + \frac{1}{w_N} \left[\frac{\tilde{\lambda}_N}{w_N} - \frac{\partial \emptyset}{\partial X_N} - \left(\frac{\partial \varphi}{\partial X_N} \right)^T \tilde{v}_N \right]. \end{aligned} \quad (28)$$

By comparing (27) with (16), we know that if two equations are equivalent, the term containing $\lambda(t_0)$ in (16) must be equal to the term containing \tilde{v}_0 and $\tilde{\lambda}_0$ in (27), i.e., $\lambda(t_0)$ depends on both \tilde{v}_0 and $\tilde{\lambda}_0$. Thus, we cannot establish the mapping relation between $\lambda(t_0)$ and $\tilde{\lambda}_0$ without additional imposed conditions. Similar arguments also hold true for the mapping relation between $\lambda(t_N)$ and $\tilde{\lambda}_N$. However, there is a feasible solution when closure conditions with feasibility tolerance are

added [35]. Here, we add a set of closure conditions to establish the mapping relationship for $\lambda(\tau_0)$ with respect to $\tilde{\lambda}_0, \tilde{v}_0$, and $\lambda(\tau_N)$ with respect to $\tilde{\lambda}_N, \tilde{v}_f$. Equations (29) and (30) are newly added closure conditions:

$$\frac{\tilde{\lambda}_0}{w_0} + \left(\frac{\partial \varphi}{\partial X_N} \right)^T \tilde{v}_0 = 0 \quad (29)$$

$$\frac{\tilde{\lambda}_N}{w_N} - \frac{\partial \theta}{\partial X_N} - \left(\frac{\partial \varphi}{\partial X_N} \right)^T \tilde{v}_f = 0. \quad (30)$$

Then, we obtain the relation mapping for $k = 0, N$:

$$\lambda(t_0) = \frac{\tilde{\lambda}_0}{w_0}, \lambda(t_N) = \frac{\tilde{\lambda}_N}{w_N}. \quad (31)$$

By considering the partial derivatives of \tilde{J} with respect to U_k for $k = 0, \dots, N$, we can similarly derive the following conditions:

$$\begin{aligned} \mu_{\text{eq}}(t_k) &= \tilde{\mu}_{\text{eq},k}/W_k \\ \mu_{\text{inq}}(t_k) &= \tilde{\mu}_{\text{inq},k}/W_k. \end{aligned} \quad (32)$$

In summary, (23), (31), and (32) establish the mapping relations between the costate variables of OCP and the KKT multipliers of the associated NLP, which provides an indirect way to examine the optimality of optimal solutions.

B. Numerical Calculation of Collocation Points

Since there is no explicit expression for the roots of $P_N(\tau)$, directly solving $P_N(\tau)$ through Legendre polynomials usually result in very low accuracy. Here, we employ the quasi-Newton algorithm to iteratively compute the roots of $P_N(\tau)$. The quasi-Newton method includes two main parts: choosing an initial guess and designing an updating law. To efficiently compute the roots of $(N+1)$ -order polynomial $g(z) = (1-z^2)P_N(z)$, the initial guess for iteration is chosen as the poles of N -order Chebyshev polynomial that has explicit mathematical representation

$$z_0 = \{\theta_k | \theta_k = \cos(\pi k/N)\} \quad (33)$$

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

$$z_{n+1} = z_n - \Delta z_n \quad (34)$$

where n denotes the step index, and Δz_n is the length of each iteration. The Legendre polynomial satisfies the following recursive relation:

$$z_n \dot{P}_N(z_n) - \dot{P}_{N-1}(z_n) = NP_N(z_n). \quad (35)$$

Then, Δz_n can be mathematically expressed as

$$\Delta z_n = \frac{z_n P_N(z_n) - P_{N-1}(z_n)}{z_n \dot{P}_N(z_n) - \dot{P}_{N-1}(z_n)} = \frac{z_n P_N(z_n) - P_{N-1}(z_n)}{NP_N(z_n)}. \quad (36)$$

Note that (36) needs to compute the values of $P_N(z_n)$ and $P_{N-1}(z_n)$, which can be calculated by the recursive relation of Legendre polynomial [33], i.e.

$$P_{m+1}(z_n) = \frac{2m+1}{m+1} z_n P_m(z_n) - \frac{m}{m+1} P_{m-1}(z_n) \quad (37)$$

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

- 1) Initial points are determined using (33).
- 2) For step n ,
 - a) recursively compute $P_N(z_n)$ using (37) with initial conditions $P_1(z_n) = 1; P_2(z_n) = z_n$;
 - b) use (34) and (36) to iteratively compute z_{n+1} .
- 3) Stop if $e = \|z_{n+1} - z_n\|_\infty < \varepsilon$.

We run this computation algorithm in MATLAB 2009a with 3.2 GHz CPU, and set $\varepsilon = 10^{-16}$. The computing time for choosing 40, 70, or 500 collocation points is 22, 31, or 35 ms, respectively. The solution efficiency satisfies the common requirements of computing resources.

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

C. Multiphase Problems

The accuracy of the PS method is high for smooth problems but unsatisfactory for nonsmooth problems. Increasing the number of collocation points can increase the accuracy in a way but also significantly reduce the computational efficiency. A multiphase strategy can be employed to apply the PS method for nonsmooth problems [36]. The basic idea is to divide the original problem into multiple phases, and apply the PS method at each phase, respectively. Additional connectivity constraints are needed to add between different phases.

For the OCP with known phase-division locations, the multiphase processing strategy is straightforward. Let $\{T_1, T_2, \dots, T_{P-1}\}$ denote the phase-division locations of a problem with P phases. Then, we can convert the original nonsmooth problem to P -phase smooth OCPs. For each phase, we can set different performance indices, state equations, and path constraints. The total performance index is the summation of all phases:

$$J = \emptyset + \sum_{p=1}^P \frac{t_f^p - t_0^p}{2} \sum_{k=0}^{N^p} w_k^p G^p(X_k^p, U_k^p, \tau_k^p). \quad (38)$$

The connectivity constraints can be added between two phases based on specific requirements. For example of a bang-bang control problem, the state variables between two phases must be continuous and the control variables are allowed to have breakpoints. The connectivity constraints can be denoted as

$$C_{\text{phase}} \left(X_f^i, U_f^i, t_f^i, X_0^j, U_0^j, t_0^j \right) = 0 \quad 1 \leq i, j \leq P. \quad (39)$$

However, the phase-division locations are usually unknown in advance for researchers. There are two strategies to deal with such issues. The first strategy is to regard phase-division locations as additional optimization variables. This strategy increases the computational complexity but is easy to implement. The second strategy is to obtain initial optimization results first by roughly applying the PS method to the original problem, then find out the fast-changing regions of the state and control inputs, and finally form a multiphase problem by taking the fast-changing regions as a new phase. The basic idea is to

increase the density of the collocation points in the fast-changing regions, thus increasing the approximation accuracy. Based on the second strategy, we develop the following algorithm.

- 1) Obtain initial optimization results by roughly applying the PS method to the original problem.
- 2) Compute the changing rate of control input $\dot{U} = DU$ using differentiation matrix and initial computation results.
- 3) Set a threshold vector \mathbf{H} for the changing rates. For any $j = 1, 2, \dots, N_u$ and $i \in [m_1, m_2] \in (0, N)$, if $\dot{U}_{ji} > H_j$, we set time domain $[\tau_{m_1-1}, \tau_{m_2+1}]$ as a fast-changing region and set this region as a new single phase.
- 4) Increase the number of collocation points in the fast-changing region to improve the computational accuracy of the constructed multiphase problem. Properly decrease the number of collocation points in the smooth region to improve the computational efficiency. Then, we can obtain the optimal solution by solving the constructed multiphase problem.

The essential idea of this method is to choose different densities of collocation points for different regions. For a smooth region, choosing sparse collocation points can achieve good computational efficiency. For a nonsmooth region, choosing dense collocation points can help improve the computational accuracy.

D. POPS

A software package is developed in MATLAB environment, called POPS, to numerically calculate the solution of generic OCPs. It consists of four modules: 1) user configuration module including the OCP description and solver parameter setup; 2) module for converting OCP into NLP; 3) module to solve NLP; and 4) module for interacting with users, including data saving, figure drawing, etc. The POPS is applicable to solve OCPs with: 1) linear and/or nonlinear functions; 2) smooth and/or nonsmooth functions; 3) free, fixed and constrained states (both initial and terminal); 4) equality and/or inequality constraints; differentiation-typed and/or integration-typed constraints; 5) Mayer, Lagrange, and Bolza-typed performance indices. 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 are willing to provide a free copy of POPS to interested readers for any noncommercial purposes. One can send the requests to the corresponding author for the latest version.

IV. APPLICATION TO AUTONOMOUS VEHICLES

In this section, we chose two typical OCPs of automated driving to demonstrate the effectiveness of the proposed framework and solver: 1) fuel-optimized ecodriving strategy in hilly road conditions; and 2) optimal path planning in an overtaking task.

A. Fuel-Optimized Automated Vehicles

For road vehicles, fuel-saving techniques have been enthusiastically studied over the past decades. One important topic

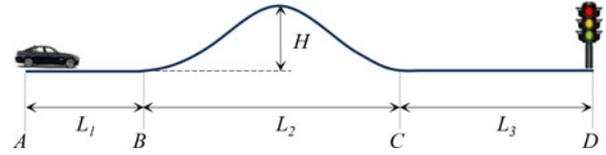


Fig. 1. Setup of fuel-optimized cruise control problem.

is the ecodriving assistance or automation. It was demonstrated that driving styles can change fuel efficiency by up to 10% even in a normal traffic flow [37]. The fuel-optimized automation aims to implement the proper control to vehicles to minimize fuel consumption. The design of such strategies is in essential to solve an OCP. This section considers a typical case study, i.e., cruise control in hilly road conditions. The vehicle considered here is a passenger car equipped with a 2.0 L gasoline engine and a continuous variable transmission (CVT).

The vehicle runs on a hilly road with both uphill and downhill. As shown in Fig. 1, 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 must be bounded as $v \in [v_{\min}, v_{\max}]$.

The longitudinal dynamic model of vehicle is given next:

$$\begin{aligned} i_g i_0 \eta_T T_e / r_w &= \delta M \dot{v} + F_R \\ F_R &= 0.5 C_D \rho_a A_v v^2 + Mg (f \cos \theta_s + \sin \theta_s) \end{aligned} \quad (40)$$

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, r_w denotes the tire effective radius, 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, and θ_s denotes the road slope. By assuming no clutch sliding, the following relationship between engine speed w_e and vehicle velocity v holds:

$$w_e = 60 v i_g i_0 / (2\pi r). \quad (41)$$

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

$$\begin{aligned} \dot{s} &= v \cos \theta_s \\ \dot{v} &= a. \end{aligned} \quad (42)$$

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

$$F_s(T_e, w_e) = \sum_{i=0}^4 \sum_{j=0}^i \beta_i \sum_{h=0}^{i+j} \left(\frac{T_e}{200} \right)^{i-j} \left(\frac{w_e}{6500} \right)^j \quad (43)$$

where β is the fitting coefficient vector. The engine BSFC is shown in Fig. 2.

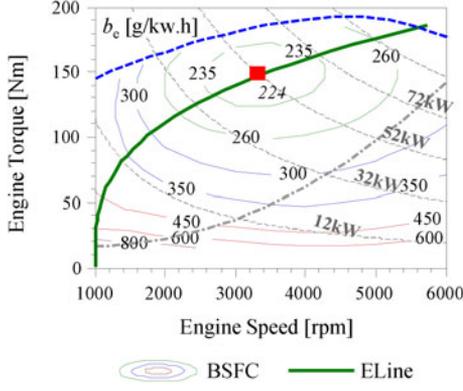


Fig. 2. BSFC map.

The optimal BSFC line (i.e., E-line) of engine is fitted as

$$T_{\text{eco}}(w_e) = k_{\text{eco}}(w_e - 1000)^\gamma \quad (44)$$

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. 2. 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.

The OCP is formulated next:

$$\min J = \int_0^{t_f} F_s + k_e (dT_e/dt)^2 dt \quad (45)$$

subject to

$$\begin{aligned} \dot{s} &= v \cos \theta_s \\ \dot{v} &= i_g i_0 \eta_T T_e / (\delta M r) - F_R / \delta M \\ w_e &= 60 v i_g i_0 / (2\pi r) \end{aligned}$$

$$\begin{aligned} T_e - k_{\text{eco}}(w_e - 1000)^\gamma &= 0 \\ s(0) &= 0 \\ s(t_f) &= L_1 + L_2 + L_3 \\ v(0) &= v_0 \\ w_{\text{emin}} \leq w_e \leq w_{\text{emax}} \\ i_{g\text{min}} &< i_g \leq i_{g\text{max}} \\ v_{\text{min}} &\leq v \leq v_{\text{max}} \\ t_{\text{min}} &\leq t_f \leq t_{\text{max}} \end{aligned}$$

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.

The POPS is applied to solve the OCP (45) with 50 collocation points. Three different initial velocities are considered, 25, 50, and 75 km/h. The optimization results are given next. For three different initial velocities, the fuel consumptions are 37.71, 27.22, and 16.75 g, respectively; the traveling time to

 TABLE I
SIMULATION PARAMETERS

Para.	Value	Para.	Value
C_D	0.316	L_1	50 m
A_p	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
$w_{\text{emin}/\text{max}}$	1000/6000 r/min	k_e	0.005
$i_{g\text{min}/\text{max}}$	0.4/2.6		

point D is 33.65, 30.17, and 23.92 s, respectively. Fig. 3 depicts the solutions of distances, velocity v , and engine torque T_e .

We can observe from Fig. 3 that on road AB , the vehicle keeps accelerating to a high speed; afterward, the engine gradually reduces its power to slow down the 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.

To compare the optimization results obtained by POPS with the operation by a driver, we design a PID controller to roughly simulate the common driving operation. The PID controller is used to keep the vehicle velocity smooth by accelerating or braking. The driving strategies obtained by POPS and PID are shown in Fig. 4.

We can observe from Fig. 4(a) and (b) that in the PID strategy the velocity decreases on the uphill road and increases on the downhill road. The velocity fluctuation is much lower than that of the POPS strategy. During the uphill road, the engine works in a noneconomy region with the maximal power output; during the downhill road, the vehicle takes braking to avoid rapid rising on velocity, which wastes a lot of kinetic energy. The total fuel of the PID strategy is around 31.26 g, while the POPS strategy only consumes 27.22 g (12.92% less). Considering the difference between final velocities, we can find that the POPS strategy produces more 35 kJ kinetic energy than the PID strategy does. So, the POPS strategy provides better driving performance than the PID strategy.

B. Trajectory Optimization in Overtaking Task

This case study presents a trajectory planning problem for automated vehicles to overtake another vehicle under mild conditions, i.e., dry road and nonaggressive driving. As shown in Fig. 5, a vehicle is required to fulfill the task of overtaking the vehicle ahead following the optimal path within the predefined trajectory region, along with the guarantee to minimize the weighted sum of squares of the steering angle and the lateral acceleration.

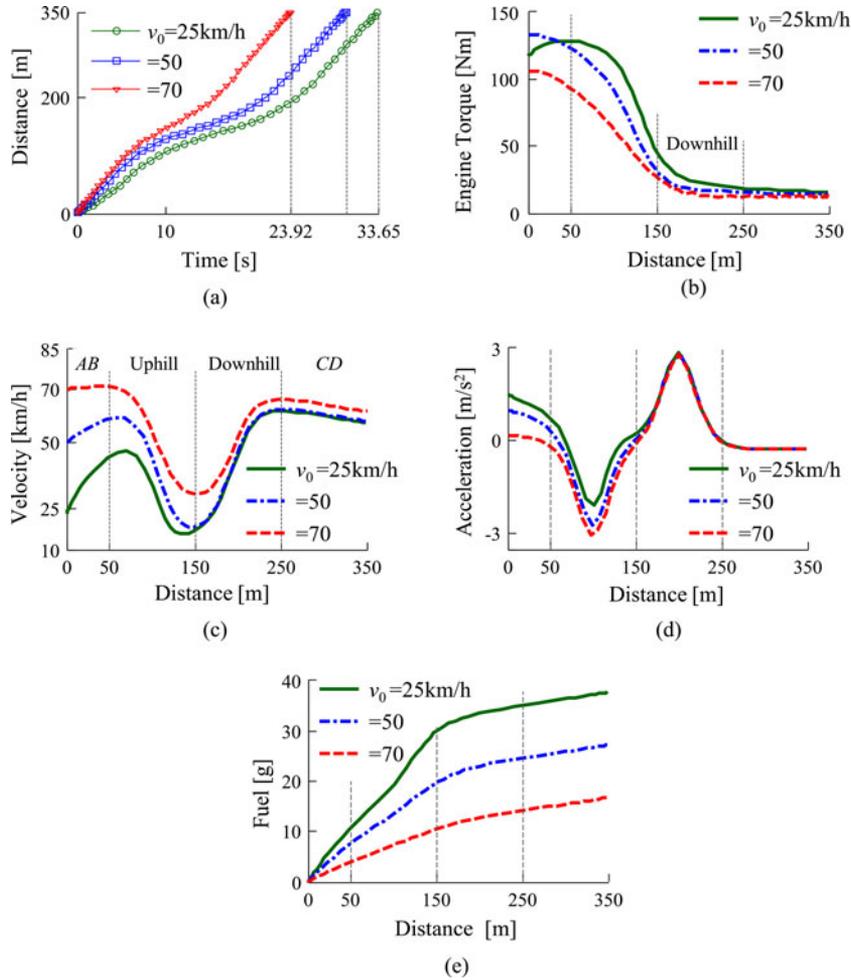


Fig. 3. POPS optimization results. (a) Driving distance versus time. (b) Engine torque versus driving distance. (c) Velocity versus driving distance. (d) Acceleration versus driving distance. (e) Fuel consumption versus driving distance.

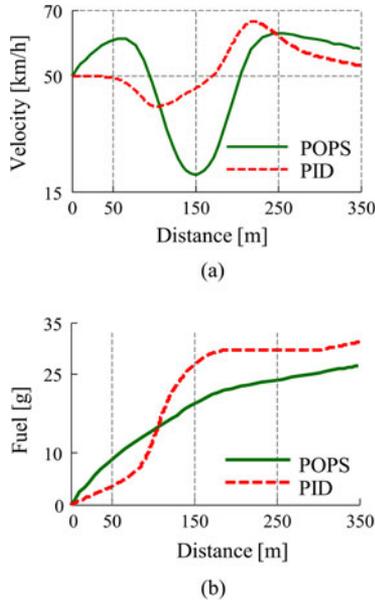


Fig. 4. Comparison of POSP optimization and PID control. (a) Comparison of velocity. (b) Comparison of fuel consumption.

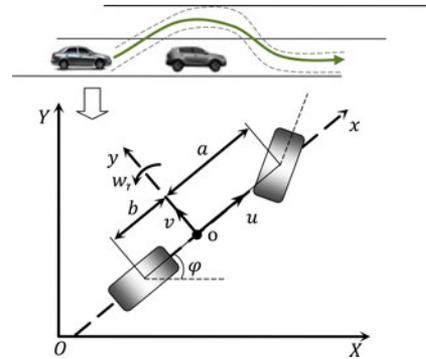


Fig. 5. Vehicle model with two degrees of freedom.

The system state equations can be obtained through the kinematic model and the lateral dynamic model of vehicles with two degrees of freedom as shown in Fig. 5. Under the assumption that the lateral tire force is proportional to the slip angle and the influence of longitudinal tire force, suspension, and slip angle on slip angle are ignored [38], the dynamics of vehicles in

coordinate system xoy is derived as

$$\begin{aligned} \dot{v} &= \frac{k_f + k_r}{mu} v + \left(\frac{ak_f - bk_r}{mu} - u \right) w_r - \frac{k_f}{mi} \delta_w \\ w_r &= \frac{ak_f - bk_r}{I_z u} v + \frac{a^2 k_f + b^2 k_r}{I_z u} w_r - \frac{ak_f}{I_z i} \delta_w \\ \dot{\varphi} &= w_r \end{aligned} \quad (46)$$

where v denotes the lateral velocity, k_f and k_r denote the cornering stiffness of two front tires and two rear tires, a and b denote the distances of the front tire and the rear tire from the center of vehicle gravity, respectively, m denotes the mass of a vehicle, u denotes the longitudinal velocity, w_r denotes yaw rate of vehicle, i denotes the transmission ratio of steering system, δ_w denotes the steering angle, I_z denotes the yaw moment of inertia of vehicle, and φ denotes yaw angle of vehicle in global axes.

To illustrate the trajectory of a vehicle, an earth coordinate system XOY is constructed, where the vehicle is located in (X, Y) . The relationship between moving trajectory and the states of vehicle can be presented as

$$\begin{aligned} \dot{X} &= u \cos \varphi - v \sin \varphi \\ \dot{Y} &= u \sin \varphi + v \cos \varphi. \end{aligned} \quad (47)$$

During the overtaking task, reducing the steering angle and the lateral acceleration can increase the performance of vehicle ride and handling. The performance function is designed as

$$\begin{aligned} J &= \int_0^{t_f} k_1 a_y^2 + k_2 \delta_w^2 dt \\ a_y &= u w_r + \dot{v} \end{aligned} \quad (48)$$

where a_y denotes as the lateral acceleration and k_1, k_2 denote the weighting coefficients of performance index. The region of overtaking trajectory T_v is demonstrated in Fig. 6(a), following the definitions:

$$\begin{aligned} T_v &= \{y \mid |y - \tilde{y}| \leq \varepsilon/2\} \\ \tilde{y} &= A_r / \left(1 + e^{-k_3(x-x_1)} \right) - A_r / \left(1 + e^{-k_4(x-x_2)} \right) \end{aligned} \quad (49)$$

where \tilde{y} denotes the center line of the designed trajectory region, ε denotes the width of a trajectory region, and A_r, k_3, k_4, x_1, x_2 denote the coefficients of the center line. The OCP for trajectory optimization is formulated as

$$\min J = \int_0^{t_f} k_1 a_y^2 + k_2 \delta_w^2 dt \quad (50)$$

subject to

$$\begin{aligned} \dot{v} &= \frac{k_f + k_r}{mu} v + \left(\frac{ak_f - bk_r}{mu} - u \right) w_r - \frac{k_f}{mi} \delta_w \\ w_r &= \frac{ak_f - bk_r}{I_z u} v + \frac{a^2 k_f + b^2 k_r}{I_z u} w_r - \frac{ak_f}{I_z i} \delta_w \\ \dot{\varphi} &= w_r \\ \dot{x} &= u \cos \varphi - v \sin \varphi \end{aligned}$$

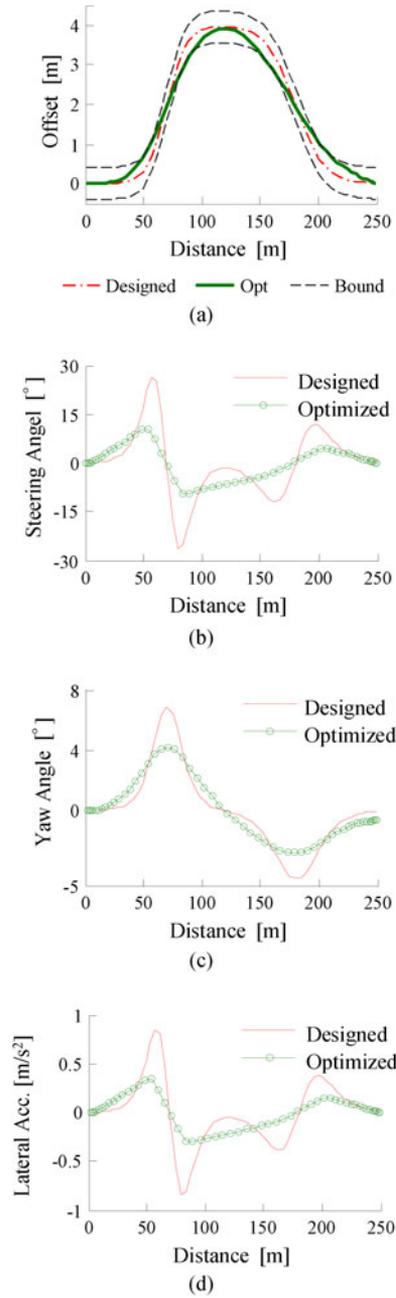


Fig. 6. POPS optimization results. (a) Vehicle trajectory. (b) Steering wheel angle. (c) Yaw angle. (d) Lateral acceleration.

$$\dot{y} = u \sin \varphi + v \cos \varphi$$

$$|y - \tilde{y}| \leq \varepsilon/2$$

$$t_{f\min} \leq t_f \leq t_{f\max}.$$

The parameters used for simulation are listed in Table II.

We explicitly use the developed solver POPS to compute the numerical solutions. The number of collocation points is set to 60. The optimal performance index is 0.0143, and the results are illustrated in Fig. 6. As shown in Fig. 6, the optimized trajectory via POPS has a relatively smoother trajectory and satisfies the boundary constraints precisely. We note that the maximum fluctuations of the steering angle decrease by more

TABLE II
SIMULATION PARAMETERS

Para	Value	Para	Value
m	1450 kg	k_1	1
a	1.2 m	k_2	0.25
b	1.5 m	E	0.8 m
I_z	1550 kg·m ²	A_r	3.5 m
k_f	-52 000 N/rad	k_3	0.12
k_r	-116 000 N/rad	k_4	0.08
u	12.5 m/s	x_1	70 m
i	20	x_2	180 m
$t_{f \max}$	30 s	ε	0.8 m
$t_{f \min}$	10 s		

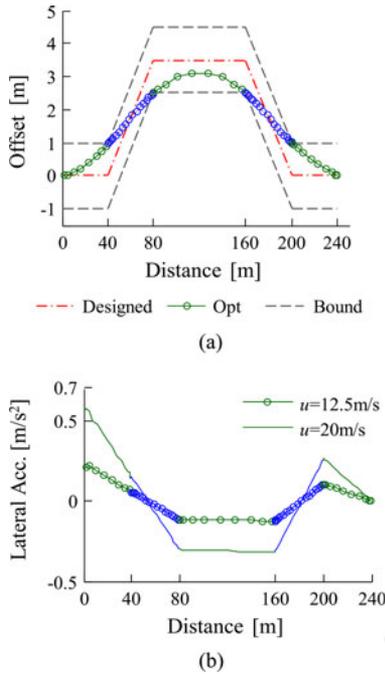


Fig. 7. POPS results for multiple phase problem. (a) Vehicle trajectory. (b) Lateral acceleration.

than 50%, no rapid fluctuations in the process, and the lateral acceleration of the vehicle is significantly reduced.

However, if the predesigned center line of trajectory is not smooth but can only be described roughly by nonsmooth functions, the POPS can still be applied to solve this problem via the multiphase preprocessing strategy. The center line of nonsmooth trajectory is described as

$$\tilde{y} = \begin{cases} 0, & x \in [0, 40] \\ 3.5/40*(x - 40), & x \in [40, 80] \\ 3.5, & x \in [80, 160] \\ 3.5 - 3.5/40*(x - 160), & x \in [160, 200] \\ 0, & x \in [200, 240] \end{cases} \quad (51)$$

which consists of five sections shown in Fig. 7(a), and ε is set to 2 m. The control variables and state variables are set to be continuous at the break points. The optimized trajectory is shown in Fig. 7(a). The optimized lateral acceleration is shown

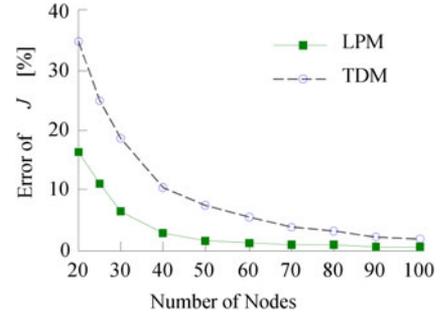


Fig. 8. Comparison of conversion rate of LPM and TDM.

in Fig. 7(b), where u is set to 12.5 and 20 m/s. This new example can illustrate the idea of multiphase strategy that dividing the original nonsmooth problem into multiple smooth problems. Actually, this kind of trajectory optimization in overtaking task is highly nonlinear. It will be much more complicated if employing the strategy to numerically solve the first-order optimality conditions.

In order to illustrate the advantages of the PS method compared to direct methods, we also solve the aforementioned problem by one TDM with following characteristics: 1) using equidistant points instead of orthogonal collocation in discretization; and 2) using composite trapezoidal quadrature rule for the integral of cost function. The convergence rate of both LPM and TDM is shown in Fig. 8, which represents the error of performance index compared to theoretical optimal solution (we use the solution of the PS method with 125 collocation points instead). Fig. 8 can clearly show that the LPM converges faster and possesses higher accuracy than TMD with the same number of collocation points does. The error of performance index is less than 1% of LPM with 60 collocation points, while the error of TDM with 60 collocation points is greater than 5.5%. Actually, other similar comparison of the PS method and TDM can reach the same conclusion that the PS method is more accurate than the traditional collocation methods [39]. The computational load is determined by the number of collocation points, initial values, and the complexity of the OCPs, e.g., the number and types of constraints. This computational framework and the developed MATLAB solver can solve most complex problems with better accuracy, but it is usually applied for offline calculation. For example, the time consumption is 2.69 and 17.04 s at 20 and 40 collocation points when initial values are set to be zero vector (in MATLAB with 3.2 GHz CPU). Greater than 95% of time is consumed to solve the converted NLP problem by a large number of iterations, which is a common issue for most optimization methods.

V. CONCLUSION

In this paper, we developed a unified computational framework based on the Legendre pseudospectral method for the purpose of accurately and efficiently calculating optimal control strategies in automotive engineering. The mapping relationship between the costate variables of OCP and the KKT multipliers of NLP is derived for the sake of checking the optimality of solutions. For engineering implementation, a quasi-Newton

iterative algorithm was designed to accurately calculate the LGL points and a multiphase preprocessing strategy was proposed to handle nonsmooth problems. A generic solver called POPS was also developed in MATLAB environment. Two case studies in the field of vehicle automation, i.e., the optimization of eco-driving strategy and the optimal path planning, were used to demonstrate the effectiveness of the developed solver. The results show that the developed framework can effectively solve complex vehicle OCPs with linear and/or nonlinear dynamics, complex constraints and different typed performance index. Compared to TDMs, this solver is more efficient and accurate for solving OCPs.

REFERENCES

[1] A. Balachandran and J. C. Gerdes, "Designing steering feel for steer-by-wire vehicles using objective measures," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 1, pp. 373–383, Feb. 2014.

[2] T. de J Mateo Sanguino and J. E. G. Ramos, "Smart host microcontroller for optimal battery charging in a solar-powered robotic vehicle," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 3, pp. 1039–1049, Jun. 2013.

[3] N. Kim, S. Cha, and H. Peng, "Optimal control of hybrid electric vehicles based on Pontryagin's minimum principle," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 5, pp. 1279–1287, Sep. 2011.

[4] X. Hu, N. Murgovski, L. M. Johannesson, and B. Egardt, "Optimal dimensioning and power management of a fuel cell/battery hybrid bus via convex programming," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 1, pp. 457–468, Feb. 2015.

[5] J. H. Crews, M. G. Mattson, and G. D. Buckner, "Multi-objective control optimization for semi-active vehicle suspensions," *J. Sound Vibration*, vol. 330, no. 23, pp. 5502–5516, 2011.

[6] W. Sun, H. Gao, and O. Kaynak, "Finite frequency H_∞ control for vehicle active suspension systems," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 2, pp. 416–422, Mar. 2011.

[7] W. Sun, Y. Zhao, J. Li, L. Zhang, and H. Gao, "Active suspension control with frequency band constraints and actuator input delay," *IEEE Trans. Ind. Electron.*, vol. 59, no. 1, pp. 530–537, Jan. 2012.

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

[9] S. E. Li, H. Peng, K. Li, and J. Wang, "Minimum fuel control strategy in automated car-following scenarios," *IEEE Trans. Veh. Technol.*, vol. 61, no. 3, pp. 998–1007, Mar. 2012.

[10] M. Kuriyama, S. Yamamoto, and M. Miyatake, "Theoretical study on eco-driving technique for an electric vehicle with dynamic programming," in *Proc. IEEE Int. Conf. Electr. Mach. Syst.*, 2010, pp. 2026–2030.

[11] M. A. S. Kamal, M. Mukai, J. Murata, and T. Kawabe, "Ecological vehicle control on roads with up-down slopes," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 783–794, Sep. 2011.

[12] F. Mensing, R. Trigui, and E. Bideaux, "Vehicle trajectory optimization for application in ECO-driving," in *Proc. IEEE Veh. Power Propulsion Conf.*, 2011, pp. 1–6.

[13] L. Nouveliere, M. Braci, L. Menhour, H. T. Luu, and S. Mammari, "Fuel consumption optimization for a city bus," in *Proc. UKACC Control Conf.*, 2008, pp. 1–6.

[14] R. F. Hartl, S. P. Sethi, and R. G. Vickson, "A survey of the maximum principles for optimal control problems with state constraints," *SIAM Rev.*, vol. 37, no. 2, pp. 181–218, 1995.

[15] D. P. Bertsekas and J. N. Tsitsiklis, "Neuro-dynamic programming: An overview," in *Proc. 34th IEEE Conf. Decision Control*, 1995, pp. 560–564.

[16] F. Fahroo and I. M. Ross, "Advances in pseudospectral methods for optimal control," in *Proc. AIAA Guid., Navigat., Control Conf. Exhibit.*, Honolulu, Hawaii, Aug. 2008, pp. 18–21.

[17] G. Elnagar, M. A. Kazemi, and M. Razzaghi, "The pseudospectral Legendre method for discretizing optimal control problems," *IEEE Trans. Automat. Control*, vol. 40, no. 10, pp. 1793–1796, Oct. 1995.

[18] F. Fahroo and I. M. Ross, "Costate estimation by a Legendre pseudospectral method," *J. Guid. Control Dyn.*, vol. 24, no. 2, pp. 270–277, 2001.

[19] Q. Gong, I. M. Ross, W. Kang, and F. Fahroo, "Connections between the covector mapping theorem and convergence of pseudospectral methods

for optimal control," *Comput. Optimization Appl.*, vol. 41, no. 3, pp. 307–335, 2008.

[20] F. Fahroo and I. M. Ross, "Pseudospectral methods for infinite-horizon nonlinear optimal control problems," *J. Guid. Control Dyn.*, vol. 31, no. 4, pp. 927–936, 2008.

[21] 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.

[22] I. M. Ross and F. Fahroo, "Convergence of pseudospectral discretizations of optimal control problems," in *Proc. 40th IEEE Conf. Decision Control*, vol. 4, 2001, pp. 3175–3177.

[23] L. N. Trefethen, *Spectral Methods in MATLAB*. Philadelphia, PA, USA: SIAM, 2000, pp. 7–40.

[24] I. M. Ross, *User's Manual for DIDO: A MATLAB Application Package for Solving Optimal Control Problems*. Vasteras, Sweden: Tomlab Optimization, Inc., 2004.

[25] A. V. Rao, D. A. Benson, C. Darby, M. A. Patterson, C. Franconin, I. Sanders, and G. T. Huntington, "Algorithm 902: A Matlab software for solving multiple-phase optimal control problems using the Gauss pseudospectral method," *ACM Trans. Math. Softw.*, vol. 37, no. 2, art. no. 22, pp. 1–39, 2010.

[26] V. M. Becerra, *PSOPT Optimal Control Solver User Manual*. Reading, U.K.: Univ. Reading, 2010.

[27] P. E. Rutquist and M. M. Edvall, *PROPT-Matlab Optimal Control Software*. Vasteras, Sweden: Tomlab Optimization, Inc., 2009.

[28] F. Fahroo and I. M. Ross, "Direct trajectory optimization by a Chebyshev pseudospectral method," *J. Guid. Control Dyn.*, vol. 25, no. 1, pp. 160–166, 2002.

[29] D. Garg, M. A. Patterson, C. Franconin, C. L. Darby, G. T. Huntington, W. W. Hager, and A. V. Rao, "Direct trajectory optimization and costate estimation of finite horizon and infinite horizon optimal control problems using a Radau pseudospectral method," *Comput. Optimization Appl.*, vol. 49, no. 2, pp. 335–358, 2011.

[30] D. Benson, "A Gauss pseudospectral transcription for optimal control," Ph.D. dissertation, Dept. Aeronautics Astronautics, MIT, Cambridge, MA, USA, 2005.

[31] I. M. Ross and M. Karpenko, "A review of pseudospectral optimal control: From theory to flight," *Annu. Rev. Control*, vol. 36, no. 2, pp. 182–197, 2012.

[32] G. T. Hunting, D. Benson, and A. V. Rao, "A comparison of accuracy and computational efficiency of three pseudospectral methods," presented at the *AIAA Guidance, Navigation, and Control Conference*, Hilton Head, SC, USA, 2007.

[33] P. G. Ciarlet and J. L. Lions, *Handbook of Numerical Analysis*, vol. 8. Houston, TX, USA: Gulf Professional Publishing, 2002.

[34] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM J. Optimization*, vol. 12, no. 4, pp. 979–1006, 2002.

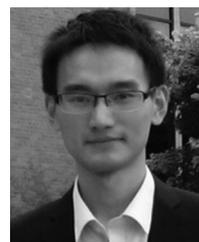
[35] Q. Gong, I. M. Ross, W. Kang, and F. Fahroo, "On the pseudospectral covector mapping theorem for nonlinear optimal control," in *Proc. 45th IEEE Conf. Decision Control*, 2006, pp. 2679–2686.

[36] I. M. Ross and F. Fahroo, "Pseudospectral knotting methods for solving nonsmooth optimal control problems," *J. Guid. Control Dyn.*, vol. 27, no. 3, pp. 397–405, 2004.

[37] J. N. Barkenbus, "Eco-driving: An overlooked climate change initiative," *Energy Policy*, vol. 38, no. 2, pp. 762–769, 2010.

[38] J. Huang and M. Tomizuka, "LTV controller design for vehicle lateral control under fault in rear sensors," *IEEE/ASME Trans. Mechatronics*, vol. 10, no. 1, pp. 1–7, Feb. 2005.

[39] G. N. Elnagar and M. A. Kazemi, "Pseudospectral Chebyshev optimal control of constrained nonlinear dynamical systems," *Comput. Optimization Appl.*, vol. 11, no. 2, pp. 195–217, 1998.

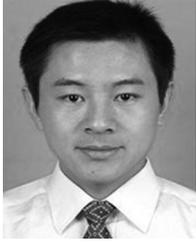


Shaobing Xu received the B.S. degree in automotive engineering from China Agricultural University, Beijing, China, in 2011, and is currently working toward the Ph.D. degree in automotive engineering at Tsinghua University, Beijing.

His research interest includes pseudospectral optimal control theory and its application on vehicle dynamics control.

Mr. Xu received awards and honors which include National Scholarship, President Scholarship, first prize of Chinese Mechanical-Design Contest,

and first prize of Beijing Advanced Mathematical Contest for College Students.

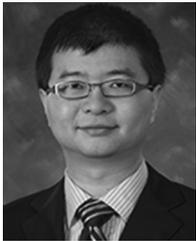


Shengbo Eben Li (M'11) received the B.S. degree in mechanical engineering from the Beijing University of Sci&Tech, Beijing, China, in 2004, and the M.S. and Ph.D. degrees in automotive engineering from Tsinghua University, Beijing, in 2006 and 2009, respectively.

From 2009 to 2011, he was a Postdoctoral Research Fellow with the University of Michigan, USA. He is currently an Assistant Professor in the Department of Automotive Engineering, Tsinghua University. His active research interests include autonomous

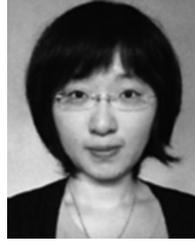
vehicles, optimal control, driver assistance systems, and lithium ion battery management. He is the author of more than 50 papers, and the coinventor of more than ten patents.

Dr. Li also served as an Associate Editor of IEEE IV Symposium (2012/2013), organization committee of FIF-ADAS forum (2013), Chinese Mechanical Engineering Society-Senior Membership. He was the recipient of Award for Science and Technology of China ITS Association (2012), Award for Technological Invention in Ministry of Education (2012), National Award for Technological Invention in China (2013), Honored Funding for Beijing Excellent Youth Researcher (2013).



Kun Deng (S'09–M'13) received the B.E. degree in automatic control and the M.S. degree in mechanical engineering from Tsinghua University, Beijing, China, in 2005 and 2007, respectively. He also received the M.S. degree in mathematics and the Ph.D. degree in mechanical engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2010 and 2012, respectively.

He is a Research Engineer at Whirlpool Corporation, Benton Harbor, MI, USA. His research interests include model reduction, control and optimization.



Sisi Li (S'13) received the B.E. degree in automation from the Beijing Technology and Business University, Beijing, China, in 2008, and the M.S. and Ph.D. degrees in electrical engineering from the New Jersey Institute of Technology (NJIT), Newark, NJ, USA, in 2010 and 2014, respectively.

She is currently a Research Associate in the Discrete Event Systems Laboratory, NJIT. Her research interests include parameter identification, control, simulation and optimization of energy-efficient systems.



Bo Cheng received the B.S. and M.S. degrees in automotive engineering from Tsinghua University, Beijing, China, in 1985 and 1988, respectively, and the Ph.D. degree in mechanical engineering from Tokyo University, Tokyo, Japan, in 1998.

He is currently a Professor at Tsinghua University, and Dean of Tsinghua University, Suzhou Automotive Research Institute. He is the author of more than 70 journal/conference papers, and the coinventor of more than 20 patents. His active research interests include autonomous vehicles, driver assistance systems, active safety, vehicular ergonomics, etc.

Dr. Cheng is the Chairman of Academic Board of SAE-Beijing, Member of Council of Chinese Ergonomics Society, Committee Member of National 863 Plan, etc.