Neurocomputing 177 (2016) 363–372

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Efficient and accurate computation of model predictive control using pseudospectral discretization

Shengbo Eben Li [a], Shaobing Xu [a], Dongsuk Kum [b,*]

[a] State Key Lab of Automotive Safety and Energy, Department of Automotive Engineering, Tsinghua University, Beijing 100084, China
[b] Cho Chun Shik Graduate School for Green Transportation, Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea

ARTICLE INFO

Article history:
Received 4 January 2015
Received in revised form
2 November 2015
Accepted 13 November 2015
Commincated by H. Zhang
Available online 27 November 2015

Keywords:
Model predictive control
Pseudospectral method
Discretization
Adaptive cruise control

ABSTRACT

The model predictive control (MPC) is implemented by repeatedly solving an open loop optimal control problem (OCP). For the real-time implementation, the OCP is often discretized with evenly spaced time grids. This evenly spaced discretization, however, is accurate only if sufficiently small sampling time is used, which leads to heavy computational load. This paper presents a method to efficiently and accurately compute the continuous-time MPC problem based on the pseudospectral discretization, which utilizes unevenly spaced collocation points. The predictive horizon is virtually doubled by augmenting a mirrored horizon such that denser collocation points can be used towards the current time step, and sparser points can be used towards the end time of predictive horizon. Then, both state and control variables are approximated by Lagrange polynomials at only a half of LGL (Legendre–Gauss–Lobatto) collocation points. This implies that high accuracy can be achieved with a much less number of collocation points, which results in much reduced computational load. Examples are used to demonstrate its advantages over the evenly spaced discretization.

1. Introduction

Model predictive control (MPC) is implemented by repeatedly solving an open-loop optimal control problem (OCP) and using the first element of the optimized control sequence as the current control action [1]. This class of controllers has received wide applications in the process industries since 1980s [2]. The main advantage of MPC is its ability to explicitly handle constraints on controls and states and achieve optimized control inside admissible sets. Generally, the plant dynamics are described in the continuous time representation by resorting to first principle equations. For the real-life implementation, however, discrete-time descriptions are required, and the continuous time problem must be converted into a discrete time problem for the computer environment [3].

The typical approach for this conversion is to discretize the plant model, constraints, and cost function at evenly spaced time points. The evenly spaced discretization approach has been widely used in the literature, e.g. Chen and Allgower [4], Muske and Badgwell [5], Qin and Badgwell [6] and Scattolini [7]. In these studies, the system behaviors between two consecutive sampling points are usually approximated by using either zero order hold (ZOH) or first order hold (FOH), which are the two lowest order approximation methods [8,9]. For reduced computational efforts, some studies used discretization methods with unevenly spaced points, e.g. moving blocking strategy [10,11]. They, however, share the same approximation methods, ZOH and FOH, which leads to significant state prediction and control errors. Hence, small sampling time is necessary in order to achieve sufficiently high levels of accuracy, which results in high computational load.

In this study, we will discretize the continuous-time MPC with unevenly spaced grid points. The unevenly spaced discretization relies on the known pseudospectral approach, which enables to use high order polynomials to approximate states and controls. The high order approximation provides more flexibility to describe the system behavior between two points, thus effectively avoiding the shortcomings of lower order counterpart. In theory, the discretized NLP converges to the OCP at a spectral rate with the number of collocation points [12].

The pseudospectral method was first applied to optimal control problems in the late 1980s. The use of Chebyshev polynomial as interpolation basis is the first method [13]. Recently, the majority have employed Lagrange polynomials as basis functions, which are often categorized into three types: Legendre–Gauss–Lobatto (LGL) method, Legendre–Gauss (LG) method, and Legendre–Gauss–Radau (LGR) method [12]. The first type uses the family of Lobatto quadrature. The collocation points of LGL, defined on the closed interval $[-1,1]$, are the roots of the

* Corresponding author.
E-mail address: dskum@kaist.ac.kr (D. Kum).

http://dx.doi.org/10.1016/j.neucom.2015.11.020
0925-2312/

derivative of N th-degree Legendre polynomial, i.e., $\dot{L}_N(t)$, together with -1 and 1. Several variants has been presented before, including [14,15] and [16]. A more general version is the Jacobi pseudospectral method presented by [17]. The second type uses the family of Gauss quadrature [18–20]. The collocation points of LG, defined on the open interval $(-1,1)$, are the zeros of N th-degree Legendre polynomial $L_N(t)$, so that the endpoints of integration interval are not included. The third type uses the family of Radau quadrature [21,22]. The collocation points of LGR, defined on the half-open interval $(-\mathbf{1},\mathbf{1}]$, are the roots of $L_{N-1}(t)+L_N(t)$, which only contains one end point. Upon cursory examination it might appear as if LGL, LG and LGR collocation are essentially similar, with only minor difference on whether contains end points. It has been shown by Garg et al. that the differences between three schemes are not merely cosmetic [31]. The LGL collocation leads to a different mathematical form as compared with either LG or LGR. As a result, LGL has different convergence properties from LG and LGR. As pointed out by Garg et al., the main differences are: (1) The differentiation matrices of LG and LGR are rectangular and full-rank, whereas that of LGL is square and singular. Therefore, LG and LGR can be written equivalently in either differential or implicit integral form while LGL does not have an equivalent implicit integral from. (2) The differentiation matrix of LGL is rank deficient. This rank-deficiency leads to a transformed adjoint system that is also rank-deficient, which can yield dual solutions that oscillate about the true solution. Conversely, LG and LGR lead to full-rank transformed adjoint systems which in turn yield approximations that converge to the true solution. (3) There is a fundamental difference on costate estimation between LGL and LGR/LG. The aforementioned oscillatory property also affects the estimation of costate estimation, which naturally attributes to the null space of the LGL transformed adjoint system. This is because that the discrete costate dynamics form a linear system of equations in terms of Lagrange multiplier. In LGL, the matrix in the equations has a null space and therefore there exists a infinite number of solutions to LGL costate dynamics. Despite the null space in LGL, many numerical examples have demonstrated that LGL has convergent approximations to state and control. The convector mapping theorem by Gong et al. [24] has pointed out that any solution of first-order optimality condition for continuous system approximately satisfies the first-order optimality condition for discrete LGL problem, and therefore the error tends to zero as $N\to\infty$. Moreover, Gong et al. [24] proposes a closure condition for selecting a good approximation to continuous costate from the infinite number of solutions.

The main study of this paper is to discretize the continuous-time MPC problem by converting the open loop OCP into NLP using the pseudospectral discretization. This unevenly spaced discretization enables more efficient and accurate implementations of MPC over evenly spaced counterpart. This paper mainly relies on the LGL collocation scheme, which makes possible to directly apply initial solution as control input and add constraints for terminal points in the predictive horizon [26]. The remainder of this paper is organized as follows: Section 2 reviews the continuous-time MPC problem. Then, its predictive horizon is designed to be augmented by a mirrored horizon in Section 3. The Legendre pseudospectral method is applied to discretize of the augmented OCP in Section 4. Section 5 renews the known covector mapping principle for the augmented problem; Illustrative examples are given in Section 6. Section 7 concludes this paper.

2. Continuous-time MPC problem

Let us consider a class of nonlinear continuous-time systems described by

$$\dot{\boldsymbol{x}}(t)=f(\boldsymbol{x}(t),\boldsymbol{u}(t)) \tag{1}$$

where $\boldsymbol{x}\in\mathbb{R}^n$ is the state vector, $\boldsymbol{u}\in\mathbb{R}^m$ is the control, $f:(\mathbb{R}^n,\mathbb{R}^m)\to\mathbb{R}^n$ is the mapping function. Note that there exists a dual pair $(\boldsymbol{x},\boldsymbol{u})\in(\mathbb{X},\mathbb{U})\in(\mathbb{R}^n,\mathbb{R}^m)$ which satisfies $f(\boldsymbol{x},\boldsymbol{u})=0$. The sets $\mathbb{X}\in\mathbb{R}^n$ and $\mathbb{U}\in\mathbb{R}^m$ are admissible box constraints for the state and input. For narrative convenience, the predictive horizon is assumed to range from $t=-1$ to $t=0$ for the sake of simplicity, where $t=-1$ is the current time. The optimal control problem for the predictive horizon (**Problem A**) is as follows.

$$\min_{\boldsymbol{u}(t),t\in[-1,0]} J=\varphi(\boldsymbol{x}(0))+\int_{-1}^{0}L(\boldsymbol{x}(t),\boldsymbol{u}(t))\mathrm{d}t \tag{2}$$

Subject to

$$\begin{aligned}&\boldsymbol{h}(\boldsymbol{x}(t),\boldsymbol{u}(t))\le 0,\\&\boldsymbol{x}(-1)=\boldsymbol{x}_0,\\&\boldsymbol{\varphi}(\boldsymbol{x}(0))\le 0,\\&\boldsymbol{x}(t)\in\mathbb{X}\in\mathbb{R}^n,\\&\boldsymbol{u}(t)\in\mathbb{U}\in\mathbb{R}^m.\end{aligned} \tag{3}$$

In **Problem A**, $t=-1$ is the initial time and $t=0$ is the final time, $L:(\mathbb{R}^n,\mathbb{R}^m)\to\mathbb{R}$ and $\phi:\mathbb{R}^n\to\mathbb{R}$ are non-negative functions of $(\boldsymbol{x},\boldsymbol{u})$ and $\boldsymbol{x}(0)$, $\boldsymbol{h}:(\mathbb{R}^n,\mathbb{R}^m)\to\mathbb{R}^q$ and $\boldsymbol{\varphi}:(\mathbb{R}^n,\mathbb{R}^m)\to\mathbb{R}^l$ reflects the constraint on the terminal state. The length of prediction horizon is 1, over which the cost function is optimized. The problem is solved numerically to obtain the optimal control trajectory $\boldsymbol{u}^*(t),t\in[-1,0]$, and the value of $\boldsymbol{u}^*(t)$ at $t=-1$ is used for the current time step control.

3. Doubled horizon: augmented by mirrored horizon

In the model predictive control defined on the interval $[-1,1]$ as shown in Fig. 1, only the first point ($t=-1$) of optimized sequence in the predictive horizon $[-1,0]$ is implemented as control input. Thus the accuracy around -1 (beginning of predictive horizon) is more important than that around $t=0$ (end of predictive horizon). Meanwhile, a point deployment such as "dense beginning and sparse end" is beneficial to enhance the accuracy of MPC. Similar technique has been used in many MPC applications, in which "move-blocking" is called [32,33]. The move-blocking scheme has two mechanisms to reduce computing complexity, of which one is to reduce the degrees of freedom by fixing some controls to be constant, and the other is to use thicker points in the beginning of predictive horizon and thinner points at the end. In contrast, the standard pseudospectral framework has the same point density at $t=-1$ and $t=+1$, which is a less efficient deployment of collocation points for MPC. In order to more effectively use LGL, the predictive horizon is virtually augmented by a mirrored horizon, which allows reducing the collocation points into half. The length of the mirrored horizon is designed to be identical to that of the predictive horizon. The states and inputs in mirrored horizon are then assumed to be symmetric to those of the predictive horizon, as illustrated in Fig. 1. The optimal control problem over the entire (predictive and mirrored) horizon is called an augmented problem, defined as **Problem B**. The mirrored strategy is designed to use denser points around initial time (i.e., $t=-1$) and sparser points around terminal time (i.e., $t=0$), which can ensure better accuracy around -1 with half-LGL collocation as compared with full LGL collocation under identical number of points. The mirrored strategy can generate similar efficiency-enhancing effect to the move-blocking technique.

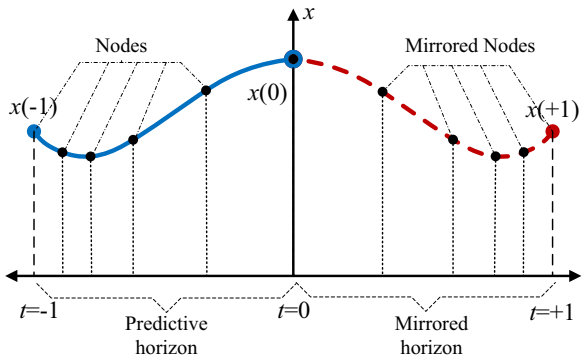


Fig. 1. Concept of augmented horizon with mirrored horizon.

The augmented optimal control problem (**Problem B**) is now defined as follows.

$$\min_{\boldsymbol{u}(t),t\in[-1,1]} J=\varphi(\boldsymbol{x}(0))+\tfrac{1}{2}\int_{-1}^{1}L(\boldsymbol{x}(t),\boldsymbol{u}(t))\mathrm{d}t \tag{4}$$

Subject to

$$\begin{aligned}&\boldsymbol{h}(\boldsymbol{x}(t),\boldsymbol{u}(t))\le 0,\\&\boldsymbol{x}(-t)=\boldsymbol{x}(+t),\\&\boldsymbol{u}(-t)=\boldsymbol{u}(+t),\\&\boldsymbol{x}(-1)=\boldsymbol{x}(+1)=\boldsymbol{x}_0\\&\boldsymbol{\varphi}(\boldsymbol{x}(0))\le 0,\\&\boldsymbol{x}(t)\in\mathbb{X}\in\mathbb{R}^n,\\&\boldsymbol{u}(t)\in\mathbb{U}\in\mathbb{R}^m.\end{aligned} \tag{5}$$

Note that both states and inputs are even functions with respect to t in the augmented problem. The plant model is modified accordingly in order to satisfy the condition that $\dot{\boldsymbol{x}}(t)$ must be an odd function in the interval $[-1,1]$

$$\begin{aligned}&\dot{\boldsymbol{x}}(t)=\bar{f}(\boldsymbol{x}(t),\boldsymbol{u}(t)),\\&\bar{f}(\boldsymbol{x},\boldsymbol{u})=\begin{cases}f(\boldsymbol{x},\boldsymbol{u}), -1\le t<0\\ 0, t=0\\ -f(\boldsymbol{x},\boldsymbol{u}), 0<t\le 1\end{cases}\end{aligned} \tag{6}$$

It is well known that the orthogonal collocation points are dense at two ends, but sparse in the middle. After augmented, the discretization around $t=-1$ are more dense than that around $t=0$, which helps to achieve more accurate feedback control even using less points. The LGL collocation scheme is adopted to convert the problem B because the LGL method is more convenient to handle initial and terminal constraints. The LGL collocation points include two endpoints at -1 and 1, making it possible to directly apply terminal constraint and implement control using the first optimized variables. Even so, LGL also suffers from some shortcomings: (1) its accuracy is a little lower than LG and LGR. If they use the same number N of collocation points, the integration accuracy of LG, LGR, and LGL are $2N-1$, $2N-2$, and $2N-3$, respectively. (2) The differentiation matrix of LGL is rank deficient, which make the accuracy of costate estimation lower than LG and LGR, and the mapping theory is also not strictly satisfied.

4. Nonlinear programming transformation using pseudospectral discretization

4.1. Basis of pseudospectral discretization method

The length of the optimal control problem is doubled, and accordingly the orthogonal collocation points should be defined to be in the form of two folds (predictive horizon and mirrored horizon). Since there exists a final constraint at $t=0$ in the original problem, an odd number of collocation points are necessary for the final constraint to be applied at $t=0$ for the augmented problem.

Let $L_{2N}(t)$ denote the Legendre polynomial of order $2N$. The orthogonal collocation points are selected to be the Legendre–Gauss–Lobatto (LGL) points, which include two parts: (1) $t_i, i=1, 2, \ldots, 2N-1$ are the zeros of $\dot{L}_{2N}(t)$, i.e. the derivative of $L_{2N}(t)$; and (2) $t_0=-1$ and $t_{2N}=1$. Given an even function $F(t)$ defined over $[-1,1]$, let us construct a $2N$-th degree Lagrange interpolating polynomial $F^{2N}(t)$. Since $F(t)$ is even, i.e. $F(t_i)=F(t_{2N-i})$, we can write $F^{2N}(t)$ as

$$\begin{aligned}&F(t)\approx F^{2N}(t)=\sum_{i=0}^{N}F(t_i)\bullet p_i(t),\\&p_i(t)=\prod_{j=0,j\ne i}^{N}\frac{t^2-t_j^2}{t_i^2-t_j^2},\end{aligned} \tag{7}$$

where $p_i(t)$ is called bi-Lagrange basis. Since it has the property $p_i(\pm t_j)=\delta_{ij}$, the isolation property holds

$$F^{2N}(t_i)=F(t_i), i=0,\ldots,N. \tag{8}$$

We derive the approximation of the integration and derivative of $F(t)$. For the integration of an even function $F(t)$, the Gauss–Lobatto quadrature formula naturally reduces the collocation points down to only $N+1$ as follow

$$\int_{-1}^{1}F(t)dt\approx\int_{-1}^{1}F^{2N}(t)dt=\sum_{i=0}^{N}w_i\bullet F(t_i) \tag{9}$$

where w_i is the weight coefficient of the Gauss–Lobatto quadrature rule, defined as

$$w_i=\begin{cases}\frac{2}{N(2N+1)\bullet[L_{2N}(t_i)]^2}, & 0\le i\le N-1\\ \frac{1}{N(2N+1)\bullet[L_{2N}(0)]^2}, & i=N\end{cases}. \tag{10}$$

The derivative of $F^{2N}(t)$ in terms of $F(t)$ at the collocation points t_i is also reduced to be

$$\dot{F}^{2N}(t_i)=\sum_{j=0}^{N}\dot{p}_j(t_i)\bullet F(t_j)=\sum_{j=0}^{N}D_{ij}\bullet F(t_j) \tag{11}$$

where $\boldsymbol{D}=(D_{ij})=\left(\dot{p}_j(t_i)\right)$ is an $(N+1)\times(N+1)$ differentiation matrix described as follows (see Appendix A for derivation):

$$D_{ij}=\begin{cases}-\frac{N(2N+1)}{2}-\frac{1}{2}, & i=j=0.\\ \frac{1}{2t_i}, & i=j=1,\ldots,N-1.\\ 0, & i=N.\\ \frac{L_{2N}(t_i)}{L_{2N}(t_N)}\bullet\frac{1}{t_i}, & i=0,\ldots,N-1, j=N.\\ \frac{L_{2N}(t_i)}{L_{2N}(t_j)}\bullet\frac{2t_i}{t_i^2-t_j^2}, & otherwise.\end{cases} \tag{12}$$

For the state and input approximation, it is well known that the polynomial interpolation of the Legendre–Gauss–Lobatto (LGL) points is more efficient than that of evenly spaced points while maintaining its accuracy [24]. This is illustrated by showing that the following interpolating error formula holds for the augmented optimal control problem.

$$\begin{aligned}&\left\|F(t)-F^{2N}(t)\right\|\le C(2N)^{\frac{1}{2}-\gamma}||F(t)||,\\&\left\|\dot{F}(t)-\dot{F}^{2N}(t)\right\|\le C(2N)^{\frac{5}{2}-\gamma}||F(t)||,\end{aligned} \tag{13}$$

where γ is the order of differentiability of $F(t)$. It is easy to know that if the function $F(t)$ is smooth enough, the convergence rate of $F^{2N}(t)$ to $F(t)$ is faster than any power of $1/2N$.

Using the Gauss–Lobatto quadrature formula with $2N+1$ LGL collocation points, the error formula is [23]

$$\int_{-1}^{1} F(t)dt - \int_{-1}^{1} F^{2N}(t)dt = \frac{-2^{4N+1}(2N+1)(2N)^3[(2N-1)!]^4}{(4N+1)[(2N)!]^4} F^{4N}(\xi) \tag{14}$$

where ξ is some value inside interval $[-1, 1]$. It is easy to know that for any integrating polynomials of degree $<4N-1$, the error of Gauss–Lobatto quadrature is exactly zero.

4.2. Transformation into nonlinear programming

Now let us discretize the augmented optimal control problem using the aforementioned pseudospectral method. We seek $2N$-th degree interpolation polynomials to approximate the state $\boldsymbol{x}(t)$ and the input $\boldsymbol{u}(t)$

$$\begin{aligned} \boldsymbol{x}(t) &\approx \boldsymbol{x}^{2N}(t) = \sum_{i=0}^{N} \boldsymbol{a}_i \bullet p_i(t), \\ \boldsymbol{u}(t) &\approx \boldsymbol{u}^{2N}(t) = \sum_{i=0}^{N} \boldsymbol{b}_i \bullet p_i(t), \end{aligned} \tag{15}$$

where the n-dimensional vector $\boldsymbol{a}_i$ and m-dimensional vector $\boldsymbol{b}_i$ are to be determined ($i=0,1,\ldots,N$). Note that $\boldsymbol{a}_i = \boldsymbol{x}^{2N}(t_i) = \boldsymbol{x}(t_i)$ and $\boldsymbol{b}_i = \boldsymbol{u}^{2N}(t_i) = \boldsymbol{u}(t_i)$. For the simplicity of notation, we write

$$\begin{aligned} &\boldsymbol{a}_{ki} = x_k(t_i), \boldsymbol{a}_i = (a_{0i}, a_{1i}, \ldots, a_{(n-1)i})^T, k = 0, \ldots, n-1. \\ &\boldsymbol{b}_{ki} = u_k(t_i), \boldsymbol{b}_i = (b_{0i}, b_{1i}, \ldots, b_{(m-1)i})^T, k = 0, \ldots, m-1. \end{aligned} \tag{16}$$

The integration in the cost function is approximated by

$$\int_{-1}^{1} L(\boldsymbol{x}, \boldsymbol{u})dt \approx \sum_{i=0}^{N} w_i \bullet L(\boldsymbol{a}_i, \boldsymbol{b}_i) \tag{17}$$

and the derivative of dynamic state constraint is approximated by

$$\dot{\boldsymbol{x}}_k^{2N}(t_i) = \sum_{j=0}^{N} D_{ij} \bullet a_{kj}, k = 0, \ldots, n-1, i = 0, \ldots, N. \tag{18}$$

In summary, the augmented optimal control problem is approximated by the following nonlinear programming (NLP) problem (**Problem $\mathbf{B^{2N}}$**), which could be solved by mature NLP tool such as SNOPT [27]:

Problem $\mathbf{B^{2N}}$: Find the $n \times (N+1)$ matrix $\boldsymbol{\alpha} = (a_{ki}) = (a_0, \ldots, a_N)$ and $m \times (N+1)$ matrix $\boldsymbol{\beta} = (b_{ki}) = (b_0, \ldots, b_N)$ that minimizes

$$\min_{\boldsymbol{\alpha} = (a_{ki}), \boldsymbol{\beta} = (b_{ki})} J^{2N} = \phi(\boldsymbol{a}_N) + \frac{1}{2} \sum_{i=0}^{N} w_i \bullet L(\boldsymbol{a}_i, \boldsymbol{b}_i) \tag{19}$$

Subject to

$$\begin{aligned} &f_k(\boldsymbol{a}_i, \boldsymbol{b}_i) = \sum_{j=0}^{N} D_{ij} a_{kj}, k = 0, \ldots, n-1, i = 0, \ldots, N-1. \\ &\boldsymbol{h}(\boldsymbol{a}_i, \boldsymbol{b}_i) < 0, i = 0, \ldots, N. \\ &\boldsymbol{a}_0 = \boldsymbol{x}_0, \\ &\boldsymbol{\varphi}(\boldsymbol{a}_N) \le 0, \\ &\boldsymbol{a}_i \in \mathbb{X} \in \mathbb{R}^n, \\ &\boldsymbol{b}_i \in \mathbb{U} \in \mathbb{R}^m. \end{aligned} \tag{20}$$

Remark (3): Note that pseudospectral discretization is generally less applicable for non-smooth problem. For a non-smooth problem, one remedy is to divide the original problem into multiple phases, and apply the pseudospectral discretization at each phase respectively. Additional connectivity constraints are add between different phases. This is usually called as a "knotting" technique [34]. In this technique, the phase-division locations are usually unknown in advance, which can be handled by two strategies. The first strategy is to take phase-division locations as additional optimization variables. This strategy increases the computational complexity but easy to implement. The second strategy is to obtain initial optimization results by roughly applying the pseudospectral discretization to original problem, identify the fast-changing regions of the state and control variables, and form a multi-phase 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. This paper does not discuss the details of knotting technique even though such an extension is easy to be added. Moreover, most MPC problems in engineering practice are often formulated into smoothing problems in a finite predictive horizon.

5. Costate estimation for augmented problem

The study of the relationship between the costate variables and the Lagrange multipliers is critical to optimality examination. As a byproduct, this section renews the known covector mapping principle (CMP) for the augmented problem. The CMP actually provides conditions under which dualization can be commuted with pseudospectral discretization [24]. Alternatively, with CMP, the costates can be determined accurately at the orthogonal collocation points by simply dividing the Karush–Kuhn–Tucker (KKT) multipliers by the quadrature weights. The derivation of CMP relies on the costate estimation, which in essence is to find the relationship between the costates in OCP and the KKT multipliers in associated NLP. For problems with full LGL points, Ross and Fahroo have proved that the KKT multipliers of NLP map linearly to the discretized costates of OCP [26]. As a result, the costates can be determined accurately by simply dividing the KKT multipliers by the LGL weights. The augmented problem in this paper uses only a half of the LGL points. This section will prove the covector mapping principle still holds for the augmented OCPs with minor changes.

We first build the relationship of transcription matrices in the conditions of half-LGL point and full LGL point. Compare with [26], the following relations hold.

$$D_{ij} = \begin{cases} \overline{D}_{ij} + \overline{D}_{i,2N-j}, & i = N \\ \overline{D}_{iN}, & i = N \\ 0, & i = N \end{cases} \tag{21}$$

$$w_i = \begin{cases} 2 \bullet \overline{w}_i = 2 \bullet \overline{w}_{2N-i}, & 0 \le i \le N-1 \\ \overline{w}_N, & i = N \end{cases} \tag{22}$$

where $\overline{D}_{ij} \in \mathbb{R}^{(2N+1)\times(2N+1)}$ and $\overline{w}_i \in \mathbb{R}^{2N+1}$ are corresponding transcription matrices for full collocation points (see [26]). The collocation is assumed to have $2N+1$ points. For **Problem B**, the Hamiltonian is defined to be

$$H(\boldsymbol{x}, \boldsymbol{u}) = \frac{1}{2} L(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\lambda}^T \overline{f}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\mu}^T \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}) \tag{23}$$

where $\boldsymbol{\lambda} \in R^n$, and $\boldsymbol{\mu} \in R^l$ are the costates associated with equality and inequality constraints. According to the first order necessary condition of optimality, the derivative of costates $\boldsymbol{\lambda}(t)$ is equal to

$$\dot{\boldsymbol{\lambda}}(t_i) = -\left(\frac{\partial H}{\partial \boldsymbol{x}}\right)(t_i) = \sum\nolimits_{j=0}^{2N} \overline{D}_{ij} \boldsymbol{\lambda}(t_j) \tag{24}$$

Therefore,

$$\frac{1}{2} \frac{\partial L}{\partial \boldsymbol{x}}(t_i) + \left(\frac{\partial \overline{f}}{\partial \boldsymbol{x}}\right)^T \boldsymbol{\lambda}(t_i) + \left(\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right)^T \boldsymbol{\mu}(t_i) = -\sum_{j=0}^{2N} \overline{D}_{ij} \boldsymbol{\lambda}(t_j) \tag{25}$$

For **Problem B^{2N}**, its optimal solution satisfies the Karush–Kuhn–Tucker (KKT) condition. The Lagrange function is defined to be

$$\sim J = \phi(a_N) + \frac{1}{2}\sum_{j=0}^{2N} \overline{w}_j L_j + \sim \boldsymbol{\nu}^T \boldsymbol{\varphi}(\boldsymbol{a}_N) + \sum_{j=0}^{2N} \left[\sim \boldsymbol{\lambda}_j^T \left(\overline{\boldsymbol{f}}_j - \dot{\boldsymbol{a}}_j \right) + \sim \boldsymbol{\mu}_j^T \boldsymbol{h}_j \right] \tag{26}$$

where $\sim\boldsymbol{\lambda}_j$, $\sim\boldsymbol{\mu}_j$, $\sim\boldsymbol{\nu}$ are the KKT multipliers of **Problem B^{2N}**. The KKT condition for the first order necessary solutions gives

$$\frac{\partial \sim J}{\partial \boldsymbol{a}_i} = 0, \frac{\partial \sim J}{\partial \boldsymbol{b}_i} = 0 \tag{27}$$

Further consider the partial derivative of $\sim J$ with respect to $\boldsymbol{a}_i (i = 1, \cdots, N-1)$

$$\frac{\partial \sim J}{\partial \boldsymbol{a}_i} = \frac{1}{2}\frac{\partial L_i}{\partial \boldsymbol{a}_i}\overline{w}_i + \left(\frac{\partial \overline{\boldsymbol{f}}_i}{\partial \boldsymbol{a}_i}\right)^T \sim \boldsymbol{\lambda}_i + \left(\frac{\partial \boldsymbol{h}_i}{\partial \boldsymbol{a}_i}\right)^T \sim \boldsymbol{\mu}_i - \frac{\partial}{\partial \boldsymbol{a}_i}\sum_{j=0}^{2N} \sim \boldsymbol{\lambda}_j^T \dot{\boldsymbol{a}}_j = 0 \tag{28}$$

where

$$\frac{\partial}{\partial \boldsymbol{a}_i}\sum_{j=0}^{2N} \sim \boldsymbol{\lambda}_j^T \dot{\boldsymbol{a}}_j = \sum_{j=0}^{2N} \overline{D}_{ji} \sim \boldsymbol{\lambda}_j = -\overline{w}_i \sum_{j=0}^{2N} \overline{D}_{ij} \frac{\sim \boldsymbol{\lambda}_j}{\overline{w}_j}. \tag{29}$$

Substituting (29) into (28), we have

$$\frac{1}{2}\frac{\partial L_i}{\partial \boldsymbol{a}_i} + \left(\frac{\partial \overline{\boldsymbol{f}}_i}{\partial \boldsymbol{a}_i}\right)^T \frac{\sim \boldsymbol{\lambda}_i}{\overline{w}_i} + \left(\frac{\partial \boldsymbol{h}_i}{\partial \boldsymbol{a}_i}\right)^T \frac{\sim \boldsymbol{\mu}_i}{\overline{w}_i} = -\sum_{j=0}^{2N} \overline{D}_{ij} \frac{\sim \boldsymbol{\lambda}_j}{\overline{w}_j}. \tag{30}$$

When (30) is identical to (25), it gives the mapping relationship between $\boldsymbol{\lambda}(t)$ and $\sim\boldsymbol{\lambda}_i$

$$\boldsymbol{\lambda}(\tau_i) = \frac{\sim \boldsymbol{\lambda}_i}{\overline{w}_i} = \frac{2 \sim \boldsymbol{\lambda}_i}{w_i}, i = 1, \cdots, N-1. \tag{31}$$

Note that (31) holds only at the collocation points when $i = 1, \cdots, N-1$. When $i = 0$ or $i = N$, we need to further consider the influence of Meyer function $\phi(\bullet)$ and final constraint $\boldsymbol{\varphi}(\bullet) = 0$. Consider the partial derivative of $\sim J$ with respect to $\boldsymbol{a}_0$, we have

$$\frac{1}{2}\frac{\partial L_0}{\partial \boldsymbol{a}_0} + \left(\frac{\partial \overline{\boldsymbol{f}}_0}{\partial \boldsymbol{a}_0}\right)^T \frac{\sim \boldsymbol{\lambda}_0}{\overline{w}_0} + \left(\frac{\partial \boldsymbol{h}_0}{\partial \boldsymbol{a}_0}\right)^T \frac{\sim \boldsymbol{\mu}_0}{\overline{w}_0} = -\sum_{j=0}^{2N} \overline{D}_{0j} \frac{\sim \boldsymbol{\lambda}_j}{\overline{w}_j} - \frac{\sim \boldsymbol{\lambda}_0}{\overline{w}_0{}^2}. \tag{32}$$

Consider the partial derivative of $\sim J$ with respect to $\boldsymbol{a}_N$, we have

$$\frac{1}{2}\frac{\partial L_N}{\partial \boldsymbol{a}_N} + \left(\frac{\partial \overline{\boldsymbol{f}}_N}{\partial \boldsymbol{a}_N}\right)^T \frac{\sim \boldsymbol{\lambda}_N}{\overline{w}_N} + \left(\frac{\partial \boldsymbol{h}_N}{\partial \boldsymbol{a}_N}\right)^T \frac{\sim \boldsymbol{\mu}_N}{\overline{w}_N} = -\sum_{j=0}^{2N} D_{Nj} \frac{\sim \boldsymbol{\lambda}_j}{\overline{w}_j} - \frac{1}{\overline{w}_N}\left(\frac{\partial \phi}{\partial \boldsymbol{a}_N} + \left(\frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{a}_N}\right)^T \sim \boldsymbol{\nu}\right). \tag{33}$$

Similarly, if (32) and (25) are identical, the right sides of two equations should have a special relationship. This implies that $\boldsymbol{\lambda}(t_N)$ depends on both $\sim\boldsymbol{\nu}$ and $\sim\boldsymbol{\lambda}_N$; otherwise, there is no way to build the mapping relationship between $\boldsymbol{\lambda}(t_N)$ and $\sim\boldsymbol{\lambda}_N$. Actually, the KKT system under LGL collocation misses the transversality conditions. Different from LG and LGR collocation, the discretization of the optimality conditions using LGL points results in an over determined system which does not match the discrete optimality conditions demonstrated from the costate endpoint conditions that are constrained by a linear combination of costate derivative condition and the transversality conditions. Despite the existence of null space in the LGL costate dynamics, a number of numerical examples has been published, which demonstrated that LGL leads to convergent approximation to state and control. Gong et al. [24] propose a closure condition for selecting a good approximation to the continuous state among the finite set of solutions to costate dynamics. Like Gong et al. [24], we can add a "closure condition" with feasibility tolerance to guarantee the transversality conditions, e.g.,

$$\left\| \sim \boldsymbol{\lambda}_0 \right\| \le \delta_D,$$
$$\left\| \sim \boldsymbol{\lambda}_N - \frac{\partial \phi}{\partial \boldsymbol{a}_N} - \left(\frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{a}_N}\right)^T \sim \boldsymbol{\nu} \right\| \le \delta_D. \tag{34}$$

where δ_D is feasibility tolerance. In addition, Gong et al. [24] shows that an asymptotically valid choice for δ_D is of the form of $\delta_D = N^{1.5-m}$, where $m \ge 4$, independent of N, but depends on the order of derivatives of optimal solution. Simultaneously, the KKT condition at $i = 0, N$ also needs to be relaxed:

$$\left\| \frac{1}{2}\frac{\partial L_0}{\partial \boldsymbol{a}_0} + \left(\frac{\partial \overline{\boldsymbol{f}}_0}{\partial \boldsymbol{a}_0}\right)^T \frac{\tilde{\boldsymbol{\lambda}}_0}{\overline{w}_0} + \left(\frac{\partial \boldsymbol{h}_0}{\partial \boldsymbol{a}_0}\right)^T \frac{\tilde{\boldsymbol{\mu}}_0}{\overline{w}_0} + \sum_{j=0}^{2N} \overline{D}_{0j} \frac{\tilde{\boldsymbol{\lambda}}_j}{\overline{w}_j} + \frac{\tilde{\boldsymbol{\lambda}}_0}{\overline{w}_0{}^2} \right\| \le \delta_D,$$

$$\left\| \frac{1}{2}\frac{\partial L_N}{\partial \boldsymbol{a}_N} + \left(\frac{\partial \overline{\boldsymbol{f}}_N}{\partial \boldsymbol{a}_N}\right)^T \frac{\tilde{\boldsymbol{\lambda}}_N}{\overline{w}_N} + \left(\frac{\partial \boldsymbol{h}_N}{\partial \boldsymbol{a}_N}\right)^T \frac{\tilde{\boldsymbol{\mu}}_N}{\overline{w}_N} + \sum_{j=0}^{2N} D_{Nj} \frac{\tilde{\boldsymbol{\lambda}}_j}{\overline{w}_j} + \frac{1}{\overline{w}_N}\left(\frac{\partial \phi}{\partial \boldsymbol{a}_N} + \left(\frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{a}_N}\right)^T \tilde{\boldsymbol{\nu}}\right) \right\| \le \delta_D. \tag{35}$$

By solving the above relaxed equations, one can get the approximated costate. Similarly, consider the partial derivative of $\sim J$ with respect to $\boldsymbol{b}_i (i = 0, \cdots, N)$, we have

$$\boldsymbol{\mu}(t_i) = \begin{cases} \frac{2\tilde{\boldsymbol{\mu}}_i}{w_i}, & i = 0, .., N-1 \\ \frac{\tilde{\boldsymbol{\mu}}_N}{w_N}, & i = N \end{cases}. \tag{36}$$

The above analysis actually builds the covector mapping principle between the KKT multiplier of **Problem B^{2N}** and the discretized costates in **Problem B**. In practice, the null space for LGL costate dynamics is often observed to be highly oscillatory. A remedy suggested by Fahroo and Ross is that the computed costate can be post-processed by using a filter to obtain a good approximation to the continuous costate [25].

6. Illustrative examples

In this section, we use two MPC examples to illustrate the superior accuracy of pseudospectral (PS) discretization over the evenly spaced (ES) discretization. The first example is a simple linear quadratic optimal control problem with control constraint, and the second example is an adaptive cruise control problem of ground vehicle. Using both PS and ES discretization, the augmented OCP in the predictive horizon is converted into NLP at each control step. The converted NLP is then numerically solved by commercial optimization software SNOPT©, in which a sparse sequential quadratic programming (SQP) algorithm is used. Note that the predictive horizon of MPC is virtually doubled by augmenting a mirrored horizon, which means only a half of LGL (Legendre–Gauss–Lobatto) collocation points are used in the predictive horizon. Therefore, the collocation points should be denser towards the current time step in the predictive horizon, and becomes sparse when going to the end time. The following simulations (e.g. Figs. 2 and 5) clearly show the use of unevenly distributed points.

Example 1: *Constrained linear quadratic control*

Example 1 is a constrained linear quadratic control problem, shown in (37). The initial state is $x(0) = 1$, and the control is bounded by $0 \le u \le 0.6$. The model predictive controller lasts 4 s and the control is implemented every $T_s = 0.2s$. (Equivalently, MPC lasts 20 steps). At each control step, the predictive horizon is set to be $T_{pred} = 3s$, and the final constraint is set to be $x(T_{pred}) = 0$. The number of half-LGL collocation points in the predictive horizon is $N+1 = 15$, in which $N = 14$ is the half-order of used Legendre

polynomial. The example 1 is to minimize

$$J = \int_0^{T_{pred}} (x^2 + u^2)\,dt \tag{37}$$

Subject to:

$$\dot{x} = -u,$$
$$0 \le u \le 0.6,$$
$$x(0) = 1.$$

Fig. 2 shows the optimized solutions of each converted NLP by the PS discretization. In Fig. 2, each line represents the optimal solutions of NLP at every 2 control steps, i.e. at the control step $t=0$ s, 0.2 s, 0.4 s, 0.6 s etc. It is easy to find that the PS discretization uses unevenly collocation points, and points are dense when close to initial time and become sparse when close to the end time of the predictive horizon. Fig. 3(a) and (b) directly compares the state and control input of MPC under the PS and ES discretization. The baseline is the solutions using points=50. When points=50, it is found that there is no obvious

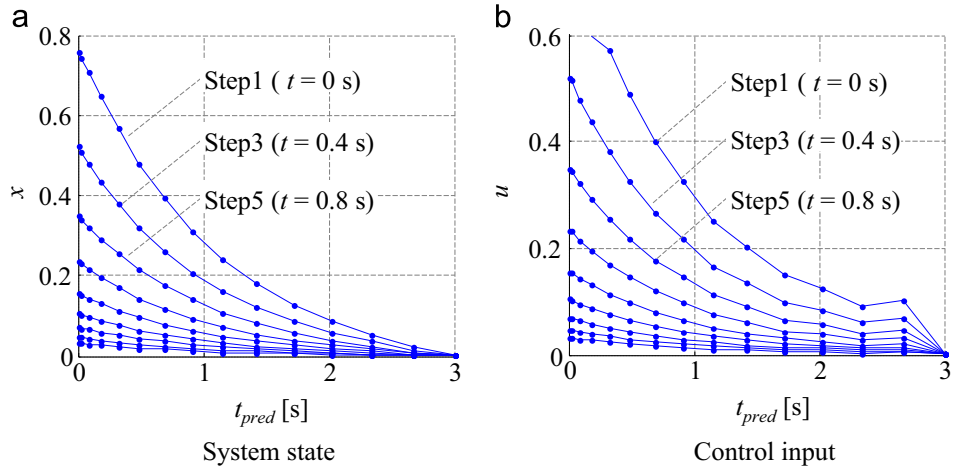


Fig. 2. Solutions of NLP discretized by PS discretization (Points=15, note that $t_{pred} \in [0, T_{pred}]$ is used to describe the time evolution in the predictive horizon).

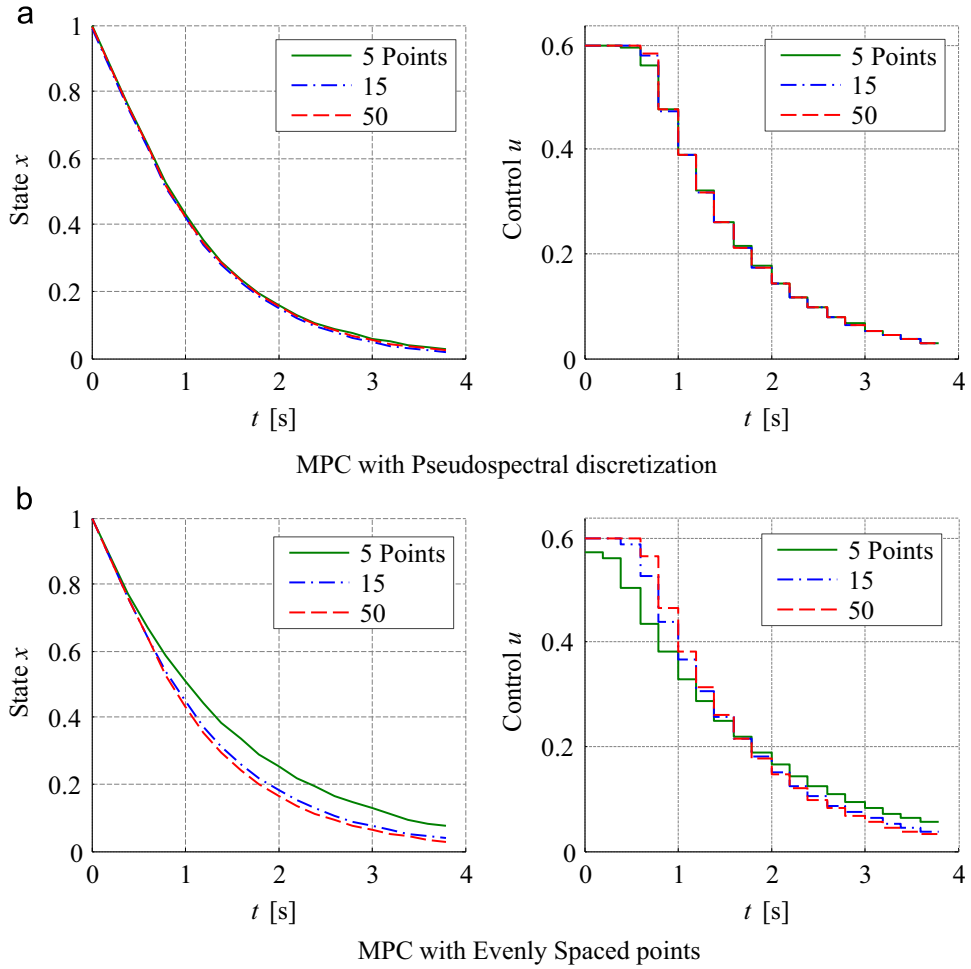


Fig. 3. State and control in MPC ($N+1=5$, 15, 50).

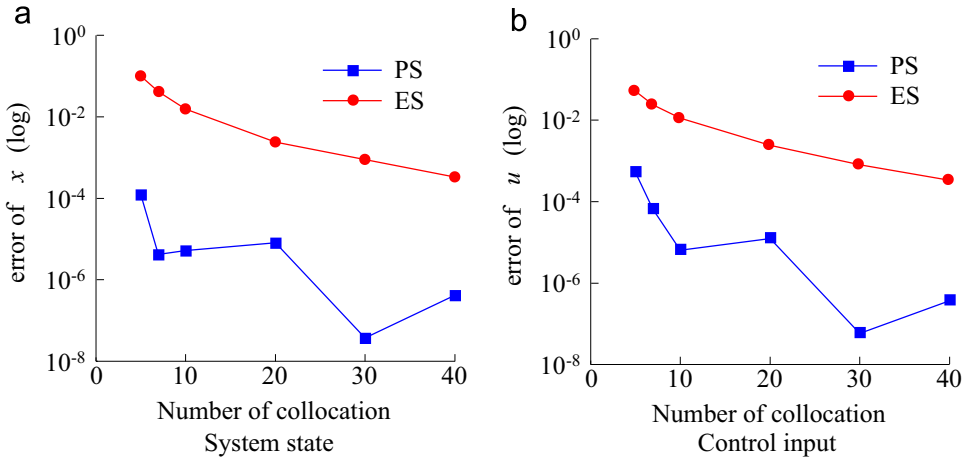


Fig. 4. Errors of system state and control input with different points.

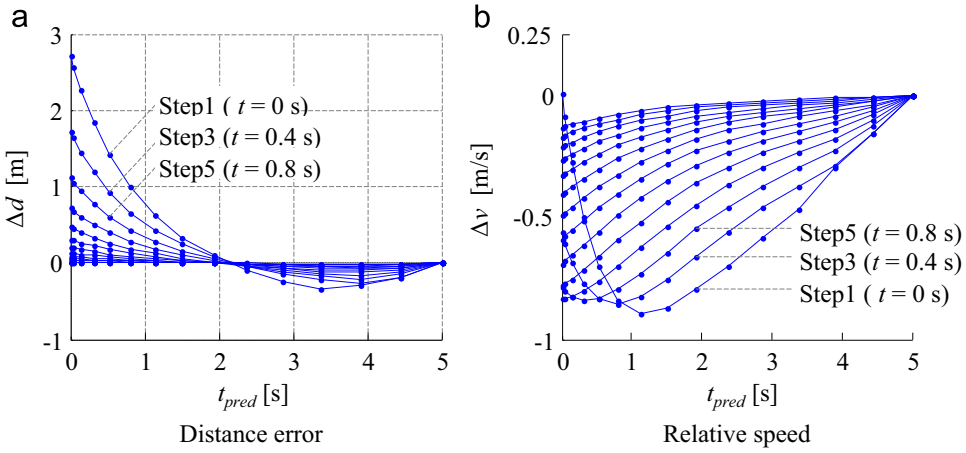


Fig. 5. Optimized solutions of NLP using PS discretization ($N=15$, note that $t_{pred} \in [0, T_{pred}]$ is used to describe the time evolution in the predictive horizon).

difference between the solutions of PS and ES, with their maximum difference in state less than 1.5×10^{-4} (error sum of square). In this case, both PS and ES discretization are accurate enough, and the accuracy of control is mainly affected by the used optimization algorithm for NLP. When using lower number of collocation number, the solutions under PS discretization is much better than the ES discretization. For the ES (Fig. 3(b)), as collocation number decreases, both states and inputs quickly go far from the baseline. For the PS (Fig. 3(a)), even using point = 5, desirable control accuracy is still guaranteed.

Fig. 4 further compares the errors of states and inputs under different collocation points (The baseline is 50 points). The error is defined to be the RMS value of errors compared to the baseline. Note that the error axis is in logarithmic scale. It is clear that the PS discretization is much more accurate than ES. In this example, the PS error using 10 points are still smaller than the ES error using 40 points.

Example 2: *Vehicular adaptive cruise control*

The main role of adaptive cruise control (ACC) of a ground vehicle is to maintain a desired distance from the prior vehicle while maintaining the desired speed. With today's growing energy crisis, one of its new properties, the fuel-saving function, is drawing widespread attention. As far as we know, the exploration of possibilities of this function goes back to the studies of Bose and Ioannou [28]. They demonstrated that the standard derivative of longitudinal acceleration was reduced by 44–52% in a single ACC car, and total fuel consumption decreased by about 8% for a mixed traffic flow with 10% ACC cars. Considering the forthcoming ACC's mass penetration into the market, this property opens up new prospects for the economic techniques of automobile industries. To design such an ACC, its fundamental is to consider not just headway-tracking capability, but also fuel economy and driver desired response comprehensively. Such a design naturally casts into the model predictive control (MPC) framework [29]. During the following example (38), the distance error Δd and relative speed Δv are included in order to maintain desirable tracking capability. The acceleration is also penalized in the cost function so as to somewhat reduce the large fuel consumption caused by urgent acceleration and deceleration [29]. The associated MPC problem is nonlinear with both equality and inequality constraints, as (38) and (39).

$$J = \int_0^{T_{pred}} \left(\Delta d^2 + \Delta v^2 + 1.8 a_f^2\right) dt \tag{38}$$

Subject to:

$$\begin{aligned}
&\Delta d = \Delta v - \tau_h a_f \\
&\Delta v = a_p - a_f \\
&a_f = \tfrac{1}{m}\left(\tfrac{T_e i_g i_0 \eta_T}{r} - mgf - \tfrac{1}{2} C_D \rho A v_f{}^2\right) \\
&\omega_e = b_{opt}\left(1 - k_{opt} T_e\right)^{-1} \\
&i_g = \tfrac{r}{i_0} \cdot \tfrac{\omega_e}{v_f} \\
&v_p = 20 + 5 \cdot \sin(0.2\pi \cdot t)\mathrm{m/s} \\
&v(0) = 1\mathrm{m/s} \\
&\Delta d(0) = 4m \\
&-20\mathrm{Nm} \le T_e \le 180\mathrm{Nm} \\
&-100\mathrm{Nm/s} \le T_e \le 100\mathrm{Nm/s}
\end{aligned} \tag{39}$$

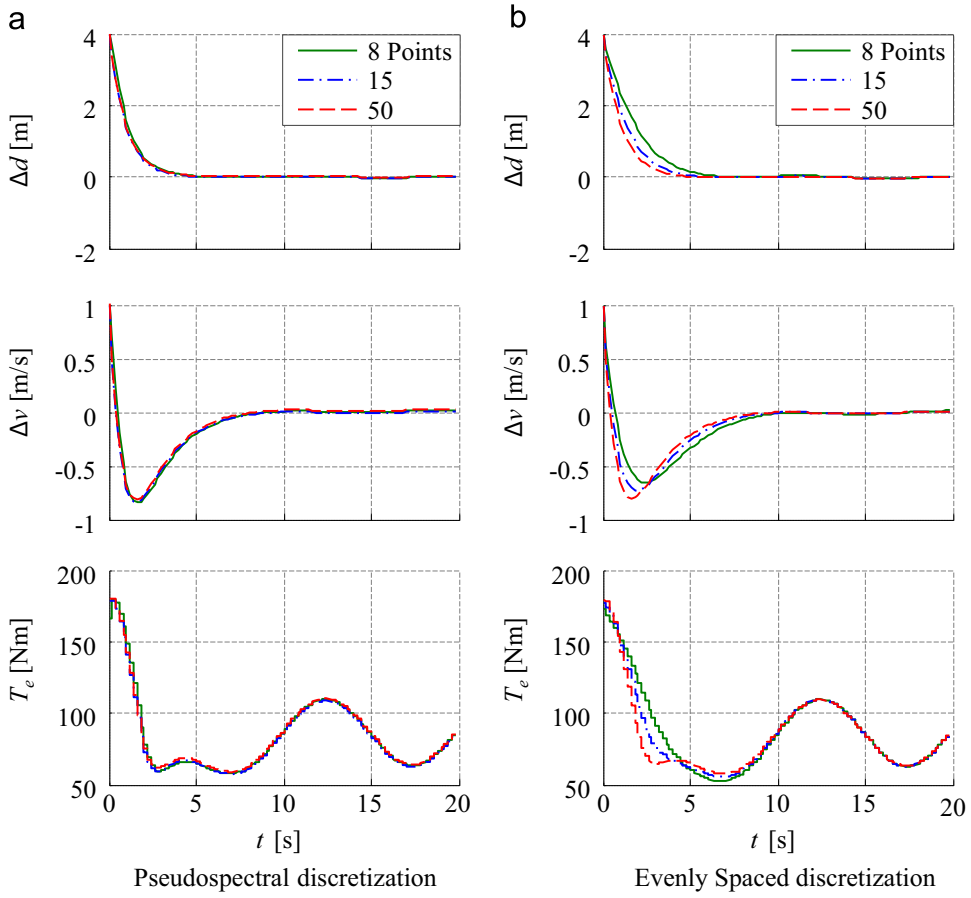


Fig. 6. State and control in MPC (N=5, 15 and 50).

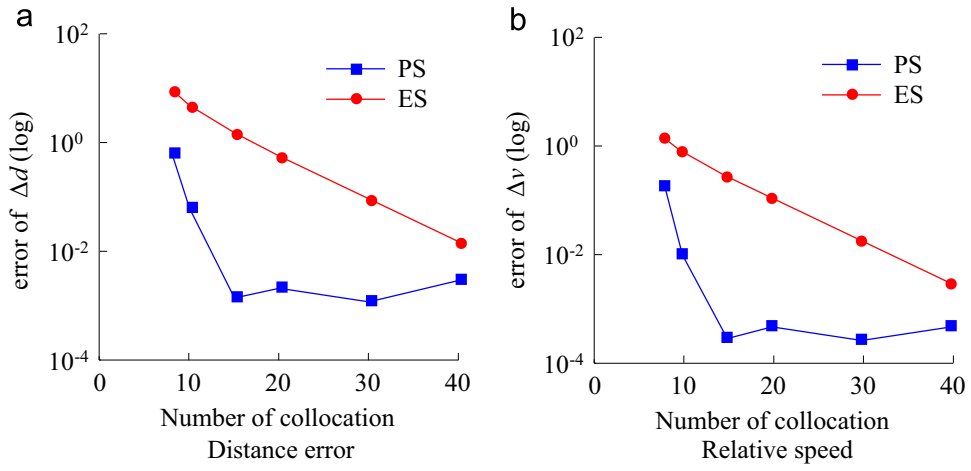


Fig. 7. Errors of states at different collocation numbers.

The model predictive controller runs totally 20 s (equivalent to 100 steps) with control interval as $T_s = 0.2s$. The initial states are $v(0) = 1m/s$ and $\Delta d(0) = 4m$. The preceding vehicle runs in a sinusoidal speed profile with average as 20 m/s and amplitude 5 m/s. The following vehicle is controlled to follow the preceding vehicle according to the aforementioned cost function and vehicle dynamics. At each control step, the predictive horizon is selected to be $T_{pred} = 5s$, and the final constraint is set to be $\Delta d(T_{pred}) = 0$ and $\Delta v(T_{pred}) = 0$. The number of half-LGL collocation points in the predictive horizon is $N+1 = 15$, in which $N = 14$ is the half-order of used Legendre polynomial.

Fig. 5 shows the optimized solutions of augmented OCP in prediction horizon (The number of collocation points is 15). Here, each line represents the optimal solutions at every 2 steps, i.e. at the control step t=0 s, 0.2 s, 0.4 s, etc. Fig. 6 compares the states and controls of two MPC at Point=8, 15 and 50 respectively. Fig. 7 shows the errors of states compared to baseline controllers (Point=50). Similar conclusion can be obtained that the PS discretization is much more accurate than the ES discretization, in particular for low collocation number. This also means that for the requirement of the same control accuracy, a much less collocation

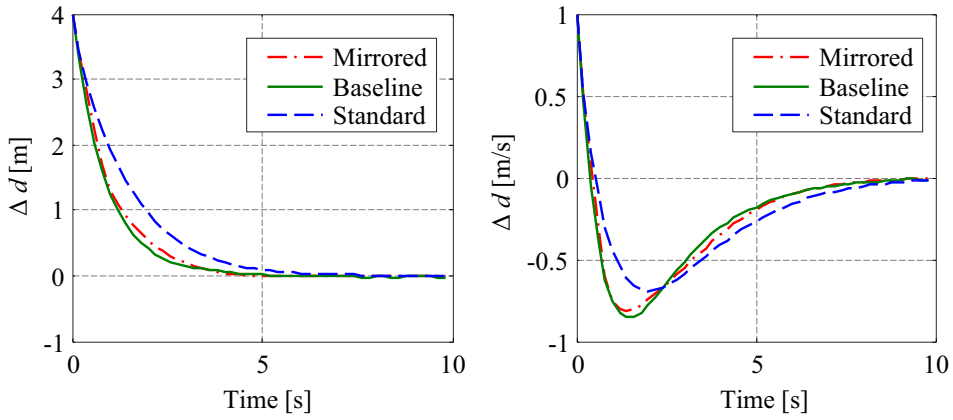


Fig. 8. Comparison of mirrored strategy and full-LGL.

number can be used in PS discretization, which will largely reduce the computational load.

To better demonstrate the benefit of mirrored strategy, a comparison simulation is given below. The number of collocation points for half-LGL and full-LGL is set to be 5, and the baseline is the result of full-LGL with 50 points. The optimal results of half-LGL, full-LGL, and the baseline are shown in Fig. 8. It is observed that the mirrored strategy is much closer to the baseline than full-LGL. This result demonstrates that the mirrored strategy can improve the accuracy of MPC with the same number of collocation points as compared with full-LGL.

7. Conclusions

This paper presents a method to efficiently and accurately compute the continuous-time MPC problem by using the pseudospectral discretization, which utilizes unevenly spaced collocation points. The conventional discretization used evenly spaced method to implement MPC. This method is easy to use, but often causes significant error because of the lower order approximation of system behavior between two sampling points. In the pseudospectral discretization, the predictive horizon of open loop OCP is virtually doubled by augmenting a mirrored horizon, which allows to reduce the orthogonal collocation points into half. Both state and control variables are approximated by Lagrange polynomials with predefined order. The integral of cost function is discretized by the Gauss–Lobatto quadrature rule. As a result, the whole discretization can guarantee to achieve high accuracy even with a much less number of collocation points. Two MPC examples are used to demonstrate its advantages over the evenly spaced discretization. The simulation result proves its effectiveness.

Acknowledgment

This research work is supported by the National Natural Science Foundation of China under Grant 51205228 and 51575293. Special thanks should be given to Prof. Bo Cheng in Tsinghua University for his suggestions on the controller design.

Appendix

Derivation of differentiation matrix

The derivation of differentiation matrix is outlined. First we write the derivative of $L_{2N}(t)$ as

$$J=\int_0^{T_{pred}}\left(\Delta d^2+\Delta v^2+1.8a_f^2\right)dt \tag{40}$$

$$L'_{2N}(t)=C_{2N}\bullet\prod_{i=1}^{2N-1}(t-t_i)=C_{2N}\bullet t\prod_{i=1}^{N-1}\left(t^2-t_i^2\right) \tag{41}$$

where C_{2N} is a number independent of interpolating points $\{t_i\}$. It is noticed that $L'_{2N}(t_i)=0$ for $i=1,\ldots,2N-1$. Now we denote

$$S(t)=C_{2N}\prod_{i=0}^{N}\left(t^2-t_i^2\right)=t\bullet\left(t^2-1\right)L'_{2N}(t) \tag{42}$$

From the property of Legendre polynomials, we have

$$\left(\left(t^2-1\right)L'_{2N}(t)\right)'=2N(2N+1)L_{2N}(t) \tag{43}$$

and we get

$$\begin{aligned}S'(t)&=\left(t^2-1\right)L'_{2N}(t)+2N(2N+1)tL_{2N}(t)\\ S''(t)&=2N(2N+1)\left(2L_{2N}(t)+tL'_{2N}(t)\right).\end{aligned} \tag{44}$$

So for all $\{t_i\}, i=0,\ldots,2N$, we have

$$\begin{aligned}S'(t_i)&=2N(2N+1)t_iL_{2N}(t_i)\\ S''(t_i)&=2N(2N+1)\left(2L_{2N}(t_i)+t_iL'_{2N}(t_i)\right).\end{aligned} \tag{45}$$

Next we derive the differentiation matrix. We present the Bi-Lagrange interpolating basis function at the node t_j by

$$p_j(t)=\prod_{i=0,i\neq j}^{N}\frac{t^2-t_i^2}{t_j^2-t_i^2}=\frac{S(t)}{S_j\bullet\left(t^2-t_j^2\right)} \tag{46}$$

where the weight S_j can be expressed by

$$S_j=\lim_{t\to t_j}\frac{S(t)}{\left(t^2-t_j^2\right)}=\begin{cases}\frac{S'(t_j)}{2t_j}, j=0,\ldots,N-1\\ \frac{S''(t_j)}{2}, j=N\end{cases} \tag{47}$$

in which the L'Hospital's rule is used [30]. By definition $D_{ij}=p'_j(t_i)$, and we know that for all off diagonal elements $(i\neq j)$,

$$D_{ij}=p'_j(t_i)=\frac{S'(t_i)}{S_j\bullet\left(t_i^2-t_j^2\right)} \tag{48}$$

By the expression of S_j we have

$$D_{ij}=\begin{cases}\frac{L_{2N}(t_i)}{L_{2N}(t_j)}\bullet\frac{2t_i}{t_i^2-t_j^2}, j=0,\ldots N-1, i\neq j\\ \frac{L_{2N}(t_i)}{L_{2N}(t_N)}\bullet\frac{1}{t_i}, j=N, i\neq j\end{cases} \tag{49}$$

For the diagonal elements, first we know that $p'_N(t_N)=p'_N(0)=0$ since $p'_N(t)$ is an odd function; then for all $i=0,\ldots N-1$,

$$D_{ii}=p'_i(t_i)=\lim_{t\to t_i}\frac{S'(t)\bullet(t^2-t_i^2)-2t\bullet S(t)}{S_i\bullet(t^2-t_i^2)^2}=\lim_{t\to t_i}\frac{S''(t)\bullet(t^2-t_i^2)-2S(t)}{S_i\bullet 2(t^2-t_i^2)\bullet 2t}=\frac{S''(t_i)}{4t_iS_i}-\frac{1}{2t_i} \tag{50}$$

in which the L'Hospital's rule is used again, and we finally get

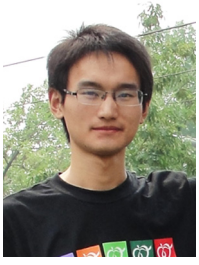
$$D_{ii}=\begin{cases}-\frac{N(2N+1)}{2}-\frac{1}{2}, i=0\\ \frac{1}{2t_i}, i=1,\ldots,N-1\\ 0, i=N\end{cases} \tag{51}$$

Thus we have derived all the components of the first order differentiation matrix in (12).

References

[1] D.Q. Mayne, J.B. Rawlings, C.V. Rao, P.O. Scokaert, Constrained model predictive control: stability and optimality, Automatica 36 (6) (2000) 789–814.
[2] M. Morari, J.H. Lee, Model predictive control: past, present and future, Comput. Chem. Eng. 23 (4) (1999) 667–682.
[3] L. Magni, R. Scattolini, Model predictive control of continuous-time nonlinear systems with piecewise constant control, IEEE Trans. Autom. Control 49 (6) (2004) 900–906.
[4] H. Chen, F. Allgöwer, A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability, Automatica 34 (10) (1998) 1205–1217.
[5] K.R. Muske, T.A. Badgwell, Disturbance modeling for offset-free linear model predictive control, J. Process Control 12 (5) (2002) 617–632.
[6] S.J. Qin, T.A. Badgwell, A survey of industrial model predictive control technology, Control Eng. Pract. 11 (7) (2003) 733–764.
[7] R. Scattolini, Architectures for distributed and hierarchical model predictive control – a review, J. Process Control 19 (5) (2009) 723–731.
[8] L. Wang, Continuous time model predictive control design using orthonormal functions, Int. J. Control 74 (16) (2001) 1588–1600.
[9] D. DeHaan, M. Guay, A real-time framework for model-predictive control of continuous-time nonlinear systems, IEEE Trans. Autom. Control 52 (11) (2007) 2047–2057.
[10] R. Cagienard, P. Grieder, E.C. Kerrigan, M. Morari, Move blocking strategies in receding horizon control, J. Process Control 17 (6) (2007) 563–570.
[11] P. Tøndel, T.A. Johansen, Complexity reduction in explicit linear model predictive control, in: Proceedings of the 15th IFAC World Congress, Spain, July 2002.
[12] G.T. Huntington, D. Benson, A.V. Rao, A comparison of accuracy and computational efficiency of three pseudospectral methods in: Proceedings of AIAA Guidance, Navigation and Control Conference and Exhibit, Hilton Head, Southern California, 2007.
[13] J. Vlassenbroeck, R. Van Dooren, A Chebyshev technique for solving nonlinear optimal control problems, IEEE Trans. Autom. Control 33 (4) (1988) 333–340.
[14] G. Elnagar, M.A. Kazemi, M. Razzaghi, The pseudospectral Legendre method for discretizing optimal control problems, IEEE Trans. Autom. Control 40 (10) (1995) 1793–1796.
[15] G.N. Elnagar, M. Razzaghi, Short communication: a collocation-type method for linear quadratic optimal control problems, Optim. Control Appl. Methods 18 (3) (1997) 227–235.
[16] F. Fahroo, I.M. Ross, Direct trajectory optimization by a Chebyshev pseudospectral method, J. Guid. Control Dyn. 25 (1) (2002) 160–166.
[17] P. Williams, Jacobi pseudospectral method for solving optimal control problems, J. Guid. Control Dyn. 27 (2) (2004) 293–297.
[18] D. Benson, A Gauss Pseudospectral Transcription for Optimal Control (Ph.D. Thesis), Massachusetts Institute of Technology, Cambridge, Massachusetts, 2005.
[19] D.A. Benson, G.T. Huntington, T.P. Thorvaldsen, A.V. Rao, Direct trajectory optimization and costate estimation via an orthogonal collocation method, J. Guid. Control Dyn. 29 (6) (2006) 1435–1440.
[20] G.T. Huntington, Advancement and Analysis of a Gauss Pseudospectral Transcription for Optimal Control Problems (Ph.D. Thesis), University of Florida, Florida, 2007.
[21] F. Fahroo, I.M. Ross, Pseudospectral methods for infinite-horizon nonlinear optimal control problems, J. Guid. Control Dyn. 31 (4) (2008) 927–936.
[22] D. Garg, M.A. Patterson, C. Francolin, et al., Direct trajectory optimization and costate estimation of finite-horizon and infinite-horizon optimal control problems using a Radau pseudospectral method, Comput. Optim. Appl. 49 (2) (2011) 335–358.
[23] P.G. Ciarlet, J.L. Lions, Handbook of Numerical Analysis, Elsevier, Gulf Professional Publishing, Amsterdam, 2002.
[24] Q. Gong, I.M. Ross, W. Kang, F. Fahroo, Connections between the covector mapping theorem and convergence of pseudospectral methods for optimal control, Comput. Optim. Appl. 41 (2008) 307–335.
[25] F. Fahroo, I.M. Ross, Costate estimation by a Legendre pseudospectral method, J. Guid. Control Dyn. 24 (2) (2001) 270–277.
[26] F. Fahroo, I.M. Ross, Advances in pseudospectral methods for optimal control, in: Proceedings of AIAA Guidance, Navigation and Control Conference and Exhibit, 2008, pp. 18–21.
[27] P.E. Gill, W. Murray, M.A. Saunders, SNOPT: an SQP algorithm for large-scale constrained optimization, SIAM J. Optim. 12 (4) (2002) 979–1006.
[28] A. Bose, P. Ioannou, Analysis of traffic flow with mixed manual and semi-automated vehicle, IEEE Trans. Intel. Transp. Syst. 4 (4) (2003) 173–188.
[29] S. Li Eben, K. Li, J. Wang, Economy-oriented vehicle adaptive cruise control with coordinating multiple objectives function, Veh. Syst. Dyn. 51 (1) (2013) 1–17.
[30] A.E. Taylor, L'Hospital's rule, Am. Math. Mon. (1952) 20–24.
[31] D. Garg, M.A. Patterson, W.W. Hager, A.V. Rao, D.A. Benson, G.T. Huntington, A unified framework for the numerical solution of optimal control problems using pseudospectral methods, Automatica 46 (11) (2010) 1843–1851.
[32] R. Cagienard, P. Grieder, E. Kerrigan, M. Morari, Move blocking strategies in receding horizon control, J. Process Control 17 (6) (2007) 563–570.
[33] S. Li Eben, Z. Jia, K. Li, B. Cheng, Fast online computation of MPC and its application to vehicular longitudinal automation, IEEE Trans. Intelli. Transp. Syst. 16 (3) (2005) 2015.
[34] I. Ross, F. Fahroo, Pseudospectral knotting methods for solving non-smooth optimal control problems, J. Guid. Control Dyn. 27 (3) (2004) 397–405.

Shengbo Eben Li received the M.S. and Ph.D. degrees from Tsinghua University in 2006 and 2009. He worked at Stanford University in 2007, University of Michigan from 2009 to 2011, and University of California, Berkeley, in 2015. He is currently an associate professor in Department of Automotive Engineering at Tsinghua University. His active research interests include autonomous vehicle control, driver modeling and driver assistance, control topics of battery, optimal control and multi-agent control, etc. He is the author of more than 80 journal/conference papers, and the co-inventor of more than 20 patents. Dr. Li 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), NSK Sino-Japan Outstanding Paper Prize in Mechanical Engineering (2013/2015), Best Student Paper Award in 2014 IEEE Intelligent Transportation System Symposium, Top 10 Distinguished Project Award of NSF China (2014), Best Paper Award in 14th ITS Asia Pacific Forum, 2015. He also served as the Associate editor of IEEE Intelligent Vehicle Symposium (2012/2013), Chairman of organization committee of China ADAS forum (2013), Guest Editor of Mathematical Problem in Engineering (2014), etc.



Shaobing Xu received the B.S. degree in automotive engineering from China Agricultural University, Beijing, China, in 2011. He is currently working on the Ph.D. degree in automotive engineering at Tsinghua University, Beijing, China. His research interest is the optimal control theory and vehicle dynamics control. His awards and honors include National Scholarship, President Scholarship, first prize of Chinese 4th Mechanical-Design Contest & first prize of 19th Advanced Mathematical Contest.

Dongsuk Kum received his Ph.D. degree in mechanical engineering from the University of Michigan, Ann Arbor, in 2010. He is currently an Assistant Professor at the Graduate School for Green Transportation in Korea Advanced Institute of Science & Technology (KAIST), and the Director of the Vehicle Dynamics and Controls (VDC) Laboratory. His research centers on the modeling, control, and design of advanced vehicular systems with particular interests in hybrid electric vehicles and autonomous vehicles. Prior to joining KAIST, Professor Kum had worked for the General Motors R&D Propulsion Systems Research Laboratory in Warren, MI as a visiting research scientist. His works at General Motors focused on advanced propulsion system technologies including hybrid electric vehicles, flywheel hybrid, and waste heat recovery systems.