SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

September 2017, Vol. 60 092201:1–092201:17
doi: 10.1007/s11432-016-0269-y

A fast algorithm for nonlinear model predictive control applied to HEV energy management systems

Lulu GUO[1,2], Bingzhao GAO[1], Yong LI[1,3] & Hong CHEN[1,2*]

[1]*State Key Laboratory of Automotive Simulation and Control, Jilin University, Changchun* 130025*, China;*
[2]*Department of Control Science and Engineering, Jilin University (Campus NanLing), Changchun* 130025*, China;*
[3]*School of Mathematics, Jilin University, Changchun* 130012*, China*

Received September 22, 2016; accepted October 31, 2016; published online March 13, 2017

Abstract This paper presents a fast algorithm for nonlinear model predictive control. In real-time implementation, a nonlinear optimal problem is often rewritten as a nonlinear programming (NLP) problem using the Euler method, which is based on dividing the prediction horizon into N steps in a given time interval. However, real-time optimization is usually limited to slow processes, since the sampling time must be sufficient to support the task's computational needs. In this study, by combining the Gauss pseudospectral method and model predictive control, a fast algorithm is proposed using fewer discrete points to transcribe an optimal control problem into an NLP problem while ensuring the same computational accuracy as traditional discretization methods. The approach is applied to the torque split control for hybrid electric vehicles (HEV) with a predefined torque demand, and its computational time is at least half that of the Euler method with the same accuracy.

Keywords energy management, hybrid electric vehicles, nonlinear model predictive control, Gauss pseudospectral method, nonlinear programming

Citation Guo L L, Gao B Z, Li Y, et al. A fast algorithm for nonlinear model predictive control applied to HEV energy management systems. Sci China Inf Sci, 2017, 60(9): 092201, doi: 10.1007/s11432-016-0269-y

1 Introduction

Nonlinear model predictive control (NMPC), an optimization-based method, is an approach to solving complex control problems that involve finding a solution at each sampling instant of a finite horizon optimal control problem. However, the solution to an on-line nonlinear optimal problem is often computationally complex and time consuming with respect to real-time NMPC implementation [1,2]. Many recent fast numerical methods have thus been proposed for solving nonlinear optimal problems.

The rigorous theoretical underpinnings of NMPC evolved within the contexts of both continuous and discrete-time systems [3]. From the theoretical point of view, numerical methods for solving continuous-time optimal control problems fall into two general categories: indirect and direct methods [4]. In indirect methods, a two-point boundary-value problem (TPBVP) is established and then solved numerically in real time. The primary advantage of indirect methods is their high accuracy; their drawbacks are small convergence radii, and the need to analytically derive the two-point boundary-value problem [5]. In

* Corresponding author (email: chenh@jlu.edu.cn)

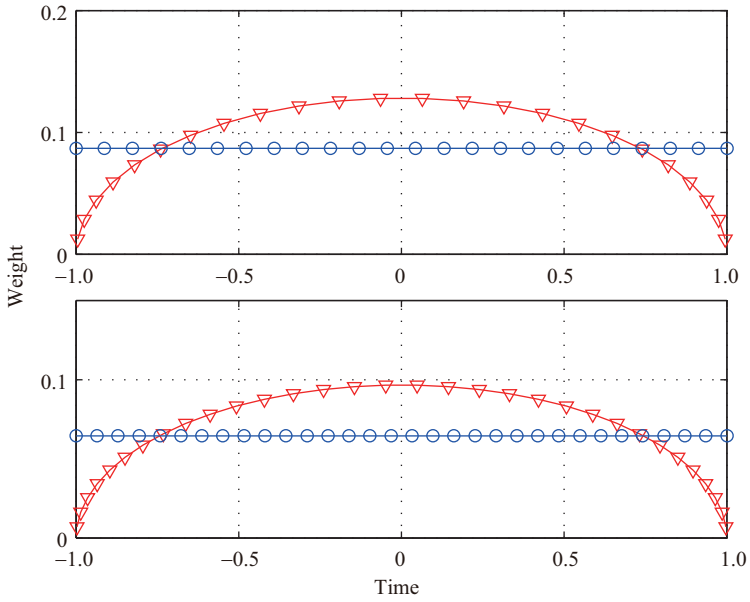

Figure 1 (Color online) Discrete points and their corresponding Euler method and Gauss pseudospectral method weights with different numbers of discrete points. The constant weight of the Euler method is defined as $A_{\text{Euler}} = 2/N$, and the Gauss weights are Legendre-Gauss (LG) points. Here, N represents the number of discrete points.

direct methods, the continuous-time optimal control problem is transcribed to a nonlinear programming (NLP) problem. The resulting NLP problem can be solved numerically by well-developed algorithms, that attempt to satisfy a set of conditions associated with the NLP problem.

There are several methods by which a continuous-time optimal control problem is transcribed into an NLP problem, one of which is the Euler method, wherein the prediction is divided into N steps over a time interval Δt. However, only slow processes should be considered for real-time implementation, as the sampling time must be sufficient to support the computational needs of the process [1]. Another approach to transcribing an optimal continuous-time control problem into an NLP problem is pseudospectral methods. Pseudospectral methods are a class of direct collocation in which global polynomials and collocation of differential-algebraic equations are used to parameterize the states and control variables [6]. One of these pseudospectral methods is the Gauss pseudospectral method (GPM), which was originally developed by [7] in effort to improve costate estimation. It was originally formulated for optimal control problems involving integral dynamic constraints, but was later adapted to be used with more common differential dynamic constraints [8], and has recently been applied to vehicle control. In [9, 10], the gear ratio is obtained by converting a switching nonlinear mixed-integer problem to a nonlinear programming problem, using a knotting technique and the Legendre pseudospectral method, which is then solved by the sequential quadratic programming (SQP) algorithm. Results show that better performance is obtained with higher computational efficiency using this method.

The GPM is based on a Gaussian quadrature formula; its basic concept is the discretization of the time horizon into non-uniform segments using Legendre-Gauss (LG) points with Gauss weights to obtain higher accuracy, as shown in Figure 1. From the figure, we can also observe that the density of these LG points at the beginning of the horizon is higher than that in the central section. Thus, by exploiting this feature and combining the GPM with NMPC, a computationally effective algorithm is proposed as a fast solution for a continuous-time optimal control problem. Its computational efficiency is evaluated by calculating a simple numerical example. The method is also applied to the energy management of a hybrid electric vehicles (HEV), solving the optimal control problem of torque split based on known torque demand, and is benchmarked with computation time and accuracy, compared with the Euler method.

The remainder of this paper is organized as follows. In Section 2, the computationally effective algorithm based on the GPM and NMPC is described. Section 3 introduces a simple numerical example, which is conducted to demonstrate the computation time and accuracy of the proposed method. An

application of the proposed method to HEV energy management is presented in Section 4, and our conclusion is given in Section 5.

2 Algorithm description

2.1 Preliminaries

In this section, we consider a class of nonlinear continuous-time systems described by

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

where $x(t) \in \mathbb{R}^m$, $u(t) \in \mathbb{R}^n$, and $p(t) \in \mathbb{R}^q$ are the states, the input variables, and the given time-dependent parameters, respectively. The control inputs at each time t are determined in order to minimize the performance index:

$$\min J = \varphi(x(t_0), x(t_\mathrm{f})) + \int_{t_0}^{t_\mathrm{f}} g(x(t), u(t), t)\,\mathrm{d}t, \tag{2}$$

where the time horizon is $t \in [t_0, t_\mathrm{f}]$, φ is the terminal constraint, and g is a cost function. Constraints are also imposed over the horizon in general, as

$$C(x(t), u(t), p(t)) \leqslant 0 \in \mathbb{R}^c, \quad \Phi(x(t_0), x(t_\mathrm{f})) = 0 \in \mathbb{R}^\phi, \tag{3}$$

where C and Φ are boundary and path constraints, respectively. Without loss of generality, the initial continuous-time NMPC problem can be reformulated as follows:

$$\min J = \varphi(x(-1), x(1)) + \frac{t_\mathrm{f} - t_0}{2} \int_{-1}^{1} g(x(\tau), u(\tau), \tau)\mathrm{d}\tau, \tag{4}$$

with constraints

$$\dot{x}(\tau) = \frac{t_\mathrm{f} - t_0}{2} f(x(\tau), u(\tau), p(\tau)), \quad C(x(\tau), u(\tau), p(\tau)) \leqslant 0 \in \mathbb{R}^c, \quad \Phi(x(-1), x(1)) = 0 \in \mathbb{R}^\phi, \tag{5}$$

where the time interval is transformed from $[t_0, t_\mathrm{f}]$ to $[-1, 1]$ via the affine transformation

$$t = \frac{t_\mathrm{f} - t_0}{2}\tau + \frac{t_\mathrm{f} + t_0}{2}, \ \tau \in [-1, 1]. \tag{6}$$

For clear expression, the notations throughout the paper are given in Table 1.

2.2 Gauss pseudospectral discretization

Considering the collocation at the N Legendre-Gauss points, noted as $\tau_1, \tau_2, \ldots, \tau_N$, and the initial point τ_0 corresponding to initial time t_0, the states $x(\tau)$ are discretized as

$$X = [x(\tau_0), x(\tau_1), x(\tau_2), \ldots, x(\tau_N)] \in \mathbb{R}^{m\times(N+1)}. \tag{7}$$

The control inputs and the given time-dependent parameters are similarly discretized to U and P as

$$\begin{aligned} U &= [u(\tau_0), u(\tau_1), u(\tau_2), \ldots, u(\tau_N)] \in \mathbb{R}^{n\times(N+1)}, \\ P &= [p(\tau_0), p(\tau_1), p(\tau_2), \ldots, p(\tau_N)] \in \mathbb{R}^{q\times(N+1)}. \end{aligned} \tag{8}$$

Then the dynamics $x(\tau)$ and $u(\tau)$ are approximated by the Lagrange interpolation polynomials at collocations points

$$x(\tau) \approx XL(\tau) \in \mathbb{R}^m, \ u(\tau) \approx UL(\tau) \in \mathbb{R}^n, \tag{9}$$

where $L(\tau)$ is the basis of Lagrange polynomials given by

$$L(\tau) = [L_0(\tau), L_1(\tau), \ldots, L_N(\tau)]^\mathrm{T} \in \mathbb{R}^{N+1}, \quad L_i(\tau) = \prod_{j=0, j\neq i}^{N} \frac{\tau - \tau_j}{\tau_i - \tau_j} \ (i = 0, 1, \ldots, N). \tag{10}$$

Table 1 Notations throughout the paper

Symbol	Description	Symbol	Description
T_{m}	Motor torque	C_{bat}	Battery capacity
P_{m}	Motor power	T_{f}	Engine torque
Δt	Sampling time	T_{dem}	Torque demand
x	State variable	ω_{f}	Engine speed
u	Control variable	ω_{m}	Motor speed
φ	Terminal constraint	$\mathrm{SoC_r}$	Battery state of charge reference
$p(t)$	Given time-dependent parameters	E_{u}	Euler method error
g	Cost function	γ	Torque split ratio
L	Basis of Lagrange polynomials	κ	Gear ratio between motor and engine
D	Differential approximation matrix	I_{g}	Transmission gear ratio
C	Boundary constraint	η_{m}	Motor efficiency
Φ	Path constraint	θ	Road slope
A_i	Gauss weights	R_{bat}	Battery internal resistance
$u^o(t)$	Optimal control input	V_{oc}	Battery open-circuit voltage
e_{u}	Error of control variable	SoC	Battery state of charge
e_{x}	Error of state	J_{g}	Objective function of GPM-MPC
$e_{\mathrm{x,max}}$	Maximal error of state	J_{e}	Objective function of Euler-MPC
$e_{\mathrm{u,max}}$	Maximal error of control variable	J_{exact}	Exact objective value
u_{exact}	Exact control results	α_{elc}	Weighting factor
x_{exact}	Exact state	m_{f}	Engine fuel rate
N_{exact}	Discrete points for exact results	u_{g}	Control inputs of GPM-MPC
u_{e}	Control inputs of Euler-MPC	N_{g}	Discrete points of GPM
N_{e}	Discrete points of Euler	J^o	Accurate cost

It follows that the differential equation of the state can be approximated by a differential operation on the Lagrange bases:

$$\dot{x}(\tau) \approx X\dot{L}(\tau) \in \mathbb{R}^m. \tag{11}$$

The derivative of each Lagrange polynomial at the LG points can be represented in a differential approximation matrix, $D = [D_1, D_2, \ldots, D_N] \in \mathbb{R}^{(N+1)\times N}$. The elements of the differential approximation matrix are determined offline as follows [5]:

$$D_k = [\dot{L}_0(\tau_k), \dot{L}_1(\tau_k), \ldots, \dot{L}_N(\tau_k)]^{\mathrm{T}} \in \mathbb{R}^{N+1}, \tag{12}$$

where $k = 1, 2, \ldots, N$. It follows that $\dot{x}(\tau_k) \approx XD_k \in \mathbb{R}^m$. Thus, the state equation in (5) is reformulated as algebraic constraints via the differential approximation matrix as follows:

$$XD_k - \frac{t_{\mathrm{f}} - t_0}{2} f(X_k, U_k, P_k) = 0,\ k = 1, 2, \ldots, N, \tag{13}$$

where $X_k \equiv x(\tau_k)$, $U_k \equiv u(\tau_k)$, and $P_k \equiv p(\tau_k)$. Regarding the terminal constraints in the preceding optimal problem, additional variables in the discretization are defined as

$$x(-1) = x(\tau_0), \quad x(1) = x(\tau_N) + \frac{t_{\mathrm{f}} - t_0}{2}(1 - \tau_N) f(X_N, U_N, P_N). \tag{14}$$

According to the LG quadrature

$$\int_{-1}^{1} f(x, \tau)\, \mathrm{d}\tau \approx \sum_{i=0}^{N} A_i f(x(\tau_i)), \tag{15}$$

an integral item in $[-1, 1]$ can be reformulated as the algebraic sum of discrete points on the integrand, where A_i is the Gauss weights. The Gauss points $\tau_1, \tau_2, \ldots, \tau_N$ are determined as the zeros of the Nth-degree Legendre polynomial [11] and the weights are the integrals of the resulting Lagrange interpolating

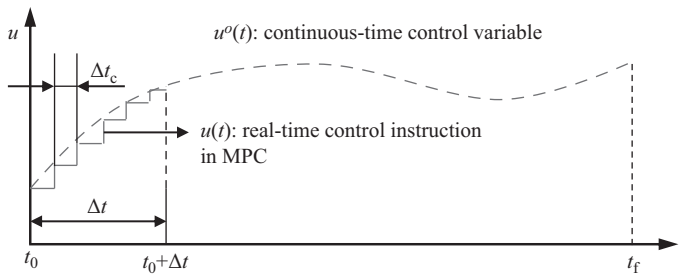

Figure 2 Control inputs of MPC using GPM in real time.

polynomials, so that [7]

$$A_i = \int_{-1}^{1} \prod_{j=0, j\neq i}^{N} \frac{\tau - \tau_j}{\tau_i - \tau_j} \mathrm{d}\tau, \ (i = 0, 1, \ldots, N). \tag{16}$$

In contrast with the Euler method, LG quadrature is a weighting summation method with a nonuniform subdivision approach; it can offer a higher accuracy with the same number of discrete points. Thus, the cost function is obtained and the initial continuous-time optimal problem can be reformulated as an NLP problem that finds the solution $Q = [X^{\mathrm{T}}, U^{\mathrm{T}}]^{\mathrm{T}}$, such that

$$\min J = \varphi(x(-1), x(1)) + \frac{t_{\mathrm{f}} - t_0}{2} \sum_{k=0}^{N} A_k g(X_k, U_k), \tag{17}$$

with constraints

$$\begin{aligned} XD_k - \tfrac{t_{\mathrm{f}} - t_0}{2} f(X_k, U_k, P_k) = 0, \quad C(X_k, U_k, P_k) \leqslant 0 \in \mathbb{R}^c, \\ \Phi(x(-1), x(1)) = 0 \in \mathbb{R}^\phi, \ k = 1, 2, \ldots, N. \end{aligned} \tag{18}$$

The resulting NLP problem is then solved numerically by well-developed algorithms.

2.3 Moving horizon strategy for online applications

As discussed above, from Figure 1, which shows a series of LG points with different average discrete intervals, we can observe that the prediction horizon is discretized with a nonuniform subdivision approach. Using the obtained discrete optimal control variable $U(\tau_k), k = 0, 1, 2, \ldots, N$, the continuous-time optimal control input, $u^o(t)(t \in [t_0, t_{\mathrm{f}}])$, can be derived by (6) and (9). In real-time applications, suppose the sampling time Δt is fixed and the interval time between two adjacent control instructions Δt_{c} is less than the sampling time. In this case, because the discrete points at the ends of the horizon are dense, it is not appropriate to use $u^o(t_0)$ directly as the control input. In general, the first several elements of the control sequence can be applied to the plant, such as

$$u(t) = u^o(t), \ t \in [t_0, t_0 + \Delta t], \tag{19}$$

as shown in Figure 2. However, if $\Delta t_{\mathrm{c}} \geqslant \Delta t$, the control instruction can be defined as

$$u(t) = u^o(t_0)\,. \tag{20}$$

Influence of the control interval Δt_{c} in control performance will be discussed in Subsection 3.2.2.

3 Numerical examples

In order to evaluate the computational efficiency and accuracy of the proposed algorithm, two numerical examples are given in this section.

3.1 Numerical example without the moving horizon strategy

As the first step in evaluating the performance of the GPM, we discuss a numerical simulation without the moving horizon strategy. Consider the following optimal control problem:

$$\begin{aligned} &\min J = \frac{1}{2}\int_0^1 (x^2(t) + u^2(t))\mathrm{d}t, \\ &\text{s.t. } \dot{x}(t) = -0.5x^2(t) + 2u(t)\left(\sqrt{x(t)} + \frac{1}{x(t)}\right), \\ &x(0) = 2,\ x(1) = 3,\ |u(t)| \leqslant 5, \end{aligned} \tag{21}$$

where $x(t) \in \mathbb{R}$ and $u(t) \in \mathbb{R}$ are the state and the control variables, respectively.

For a computational efficiency comparison, the preceding optimal control problem is separately reformulated as an NLP problem using the GPM and the Euler method using different numbers of discrete points. The resulting NLP problem is then solved by the SQP algorithm. The time horizon is transformed back to $[t_0, t_\mathrm{f}]$ by (6) after obtaining the optimal results, and the state and control variable errors are defined as

$$\begin{aligned} &e_\mathrm{u}(t_k) = |u(t_k) - u_\mathrm{exact}(t_k)|,\ e_\mathrm{x}(t_k) = |x(t_k) - x_\mathrm{exact}(t_k)|, \\ &e_\mathrm{u,max} = \max\{e_\mathrm{u}(t_k)\},\ e_\mathrm{x,max} = \max\{e_\mathrm{x}(t_k)\},\ k = 0, 1, \ldots, N, \end{aligned} \tag{22}$$

where $u_\mathrm{exact}(t_k)$ and $x_\mathrm{exact}(t_k)$ represent the exact results, which are obtained by a large number of discrete points, $N_\mathrm{exact} = 200$, using the Euler method.

Figure 3 shows the state and control input trajectories during the time horizon. Figure 4 represents the maximum error of the control variables and the states using the GPM and the Euler method with different numbers of discrete points. From these results, we can see that with the same number of discrete points, the accuracy of the GPM is much higher than that using the Euler method; by increasing the number of collection points, the maximum error in the GPM decreases rapidly to a feasible level. This is expected because in the GPM, the integral item (cost function of the optimal problem) is expressed as the algebraic sum of the discrete points using a weighting summation method, which represents a higher accuracy in comparison to linearization methods, according to LG quadrature. Furthermore, the discretization of the state equation using polynomial method in the GPM shows better precision, especially for continuous-time systems. Thus, in real time, under the same accuracy requirements, the number of discrete points can be reduced, which contributes to the reduction of computation time.

From the results shown in Figure 3, we can also observe that the early errors are much lower than the maximum ones. In real-time applications, for most numerical algorithms, only the first element or the first several elements of the derived control sequence is applied to the plant within one control interval. Thus, influence on real-time control performance depends on the errors at the beginning of the horizon. For more detailed observation, errors in one-tenth of the horizon are chosen, and are redefined as

$$e'_\mathrm{u,max} = \max\{e_\mathrm{u}(t_k)\},\ e'_\mathrm{x,max} = \max\{e_\mathrm{x}(t_k)\},\ k = 0, 1, \ldots, h, \tag{23}$$

with h determined by

$$t_{h-1} < \frac{1}{10}(t_\mathrm{f} - t_0) \leqslant t_h\,. \tag{24}$$

Figure 5 shows the redefined maximum errors $e'_\mathrm{u,max}$ using the GPM and Euler method under different numbers of discrete points. By comparing the results in Figures 4 and 5, we can see that the improvement potential of the GPM in terms of computational accuracy is larger when considering only the early errors. This will be further discussed in the next subsection.

3.2 Numerical example with the moving horizon strategy

3.2.1 *Comparison of the two methods*

This subsection deals with the evaluation of the computational performance of the proposed method by combining the GPM with a moving horizon strategy. To better demonstrate the benefit of the proposed

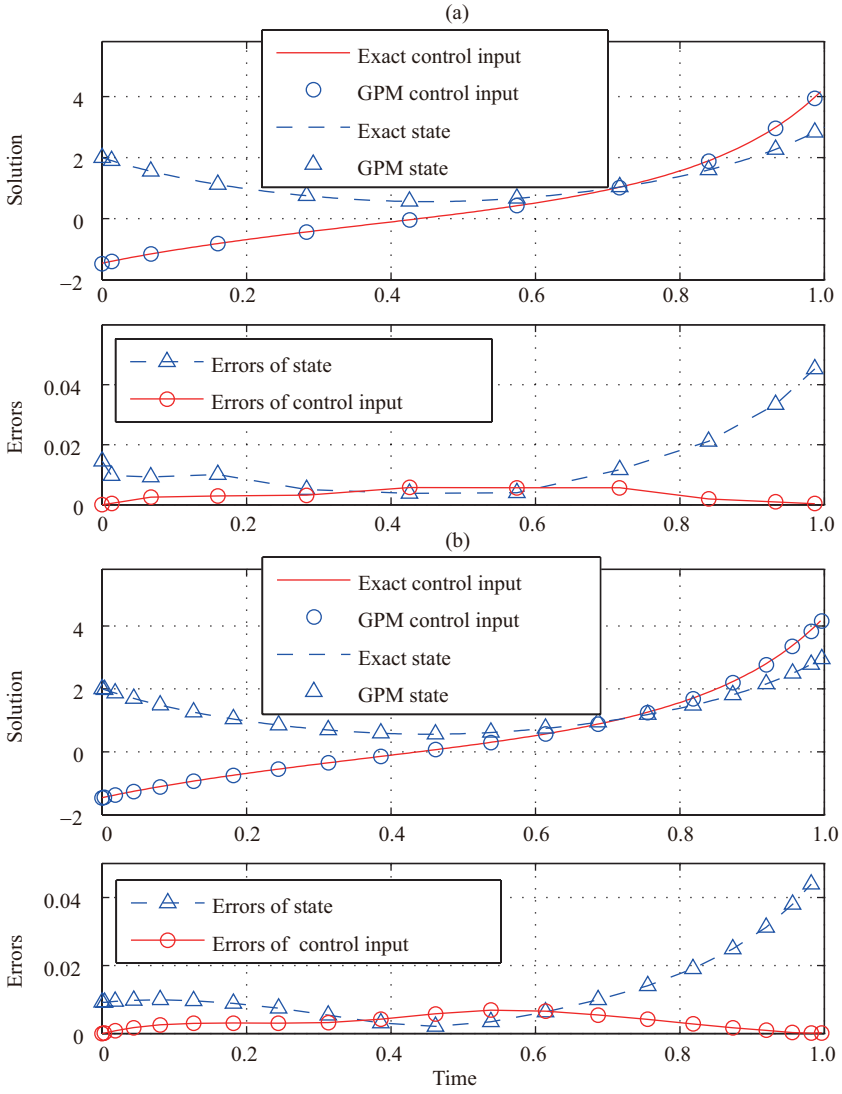

Figure 3 (Color online) State, control input, and error trajectories during the time horizon using the GPM. From top to bottom, the figures show the results using different LG points, as (a) N=10, and (b) N=20, respectively.

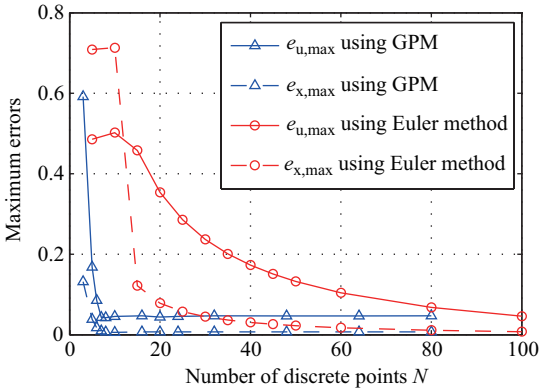

Figure 4 (Color online) The maximum error of the control variables and states using the GPM and Euler method with different numbers of discrete points.

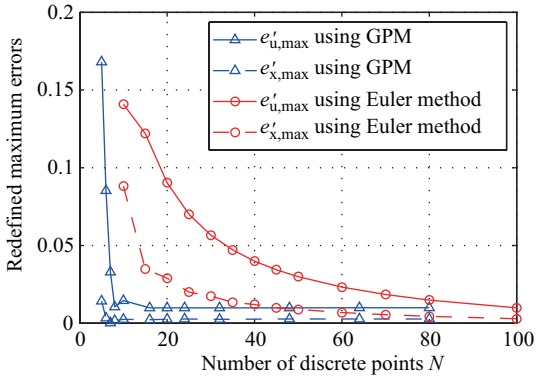

Figure 5 (Color online) The redefined maximum errors of the control variables and states using the GPM and Euler method with different numbers of discrete points.

Table 2 Errors using GPM-MPC and Euler-MPC

No.	GPM		Euler	
	N_{g}	$E(\times 10^{-5})$	N_{e}	$E(\times 10^{-5})$
1	6	498	15	236
2	8	183	20	136
3	10	71	25	79
4	16	18	30	50
5	20	11	40	44
6	24	10	50	35
7	32	12	80	25

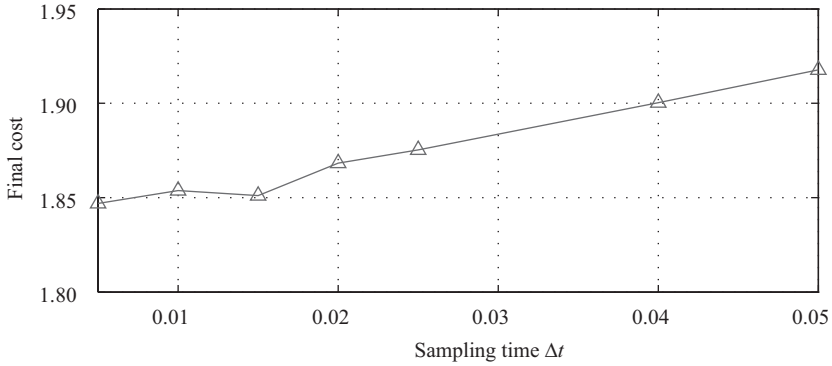

Figure 6 Final cost using different control intervals and sampling time.

strategy, a comparison simulation is given below. We consider the following NMPC problem:

$$\min J = \frac{1}{2}\int_{t}^{t+T}(x^2(t') + u^2(t'))\mathrm{d}t', \tag{25}$$

subject to the state equation in (21), $x(t+T) = 3$ and $x(t,0) = 2$, wherein the receding horizon T is defined as $T = 1 - t$. For clarity, the GPM and Euler method in the MPC framework are respectively noted as GPM-MPC and Euler-MPC, and their corresponding control inputs are defined as u_{g} and u_{e}. Additionally, the error is defined as

$$E = |J - J^o|, \tag{26}$$

with J^o defined as the accurate cost.

Table 2 summarizes the GPM-MPC and Euler-MPC errors using different numbers of discrete points, marked as N_{g} and N_{e}, respectively. From these results, we can see that under the same accuracy requirement, the computational efficiency of GPM-MPC is much higher than that of Euler-MPC. Thus, for a continuous-time control problem, the potential of further computational efficiency improvement can be realized by combining the GPM with a moving horizon strategy.

3.2.2 *Influence of the control interval* Δt_{c}

As discussed above, after obtaining the continuous-time optimal control input $u^o(t)$, the control instruction can be determined by (19) if $\Delta t_{\mathrm{c}} \leqslant \Delta t$, which means that a difference occurs between the control interval and sampling time. To observe the influence of Δt_{c} on control performance, we conducted a simple comparison simulation. The optimal problem is the same as (25) in Subsection 3.2.1 and the number of LG points is $N_{\mathrm{g}} = 20$.

Figure 6 shows the final cost results under different sampling times and control intervals. From these results, we can see that the final cost increases with the sampling time. Thus, in real-time applications, if the control interval is smaller than the sampling time, the first several optimal control inputs can be applied for better performance. This may be possible when considering sensor capacity limits. However, for most controls, $u^o(t_0)$ is applied to the system because the sampling time is sufficiently small.

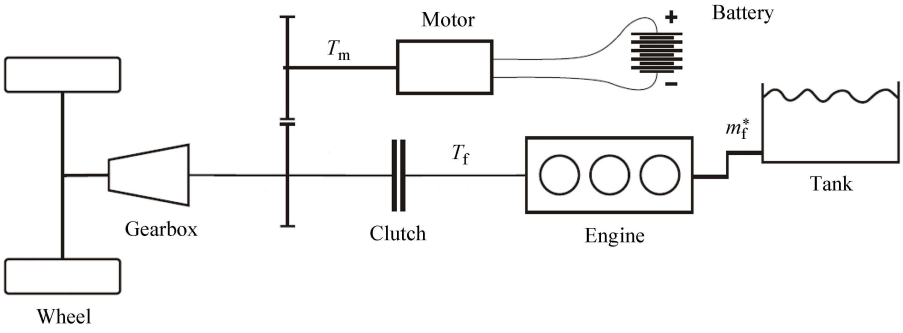

Figure 7 Topology of the parallel hybrid electric vehicle powertrain [14].

Table 3 Vehicle parameters

Symbol	Description	Value
M	Vehicle mass	1800 (kg)
ρ	Air density	1.205 (kg/m^3)
A_{f}	Face area	2.08 (m^2)
c_{d}	Coefficient of air resistance	0.36
f	Coefficient of rolling resistance	0.011
η_{t}	Drive train total efficiency	0.96
r_{w}	Dynamic tire radius	0.33 (m)
C_{bat}	Battery charge capacity	20 (Ah)

4 HEV energy management

Hybrid electric vehicles (HEVs), which are a more viable option for reducing energy consumption and emissions than conventional vehicles, have become popular in recent years [12]. To fully utilize the capability of regenerative braking and the potential of optimizing the operation of its energy sources, splitting torque between the two sources without affecting vehicle speed has become a critical issue in HEV control [13]. In this section, using the GPM-MPC, a real-time solution for determining the optimal torque-split ratio of an HEV is proposed. Figure 7 shows the system under consideration, a passenger vehicle with a parallel hybrid electric drivetrain. The drivetrain comprises an internal combustion engine (ICE) and an electric capacity device, the parameters of which are shown in Table 3.

4.1 Methodology and problem formulation

Many power flow paths can be created within the parallel HEV powertrain in response to the positive torque demand. Each of these flow paths can be identified schematically by vehicle modes and numerically by the torque-split ratio $\gamma(t)$, which is defined as the ratio of the torque contribution of the electric system to the total torque demand [15], as follows:

- $\gamma(t) = 0$, pure thermal mode: the torque demand is given by the engine only.
- $0 < \gamma(t) < 1$, hybrid mode: both the engine and the motor provide the torque, denoted as T_{f} and T_{m}, respectively.
- $\gamma(t) < 0$, recharging mode: the electric motor is engaged with wheels as a generator producing electricity.

Pure thermal mode is preferred when the engine operates in its peak-efficiency region. Pure electric mode is selected when the engine operates in a low-efficiency zone, such as at low speeds. In hybrid mode, both the engine and the motor are used. When the state of the battery charge is low, the engine produces excess energy to recharge the battery efficiently; conversely, when the state of the battery charge is high, pure electric mode is used. It should be noted that during deceleration, the clutch disengages the engine from the driveline. The electric motor alone remains engaged to recuperate energy and thus acts as a generator. Thus in this case, the only control variable is the motor torque (with respect to power source),

which can be calculated by the powertrain ratio and the brake force required. Then, a brake recovery control system can be used. This calculation is outside the scope of this study.

For the torque-split optimization, the objective is to find the optimal torque ratio that reduces energy consumption and maintains state-of-charge (SoC) stability. In equations, we find the control policy $u = \gamma(t)$, $t' \in [t, t+T]$, such that [16]

$$\min J = \int_t^{t+T} m_{\mathrm{f}}(t') + \alpha_{\mathrm{elc}}(\mathrm{SoC}(t') - \mathrm{SoC_r})^2 \mathrm{d}t', \tag{27}$$

subject to

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{SoC}(t) = -\frac{V_{\mathrm{oc}} - \sqrt{V_{\mathrm{oc}}^2 - 4P_{\mathrm{m}}(t)R_{\mathrm{bat}}}}{2C_{\mathrm{bat}}R_{\mathrm{bat}}}, \tag{28}$$

where $m_{\mathrm{f}}(t)$ is the fuel rate of the engine; α_{elc} is a weighting factor; $\mathrm{SoC_r}$ is the reference battery state-of-charge, $P_{\mathrm{m}}(t)$ is the motor power; and V_{oc}, R_{bat} and C_{bat} are the battery open-circuit voltage, internal resistance, and capacity, respectively. It should be noted that positive battery power $P_{\mathrm{m}}(t) > 0$ indicates that the battery is discharging and negative battery power $P_{\mathrm{m}}(t) < 0$ indicates that the battery is charging.

In general, the motor power is a function of the motor torque $T_{\mathrm{m}}(t)$ and speed $\omega_{\mathrm{m}}(t)$, as

$$\begin{aligned} P_{\mathrm{m}}(t) &= T_{\mathrm{m}}(t)\omega_{\mathrm{m}}(t)\eta_{\mathrm{m}}^{-\mathrm{sign}(T_{\mathrm{m}}(t))}, \\ T_{\mathrm{m}}(t) &= \gamma(t)T_{\mathrm{dem}}(t), \end{aligned} \tag{29}$$

where η_{m}, the efficiency of the motor, is used to calculate the power loss of the motor as a function of the corresponding torque and speed and T_{dem} is the torque demand from the driver. The engine fuel rate $m_{\mathrm{f}}(t)$, is often expressed as a map, extracted rom the engine, as

$$m_{\mathrm{f}}(t) = \Psi(T_{\mathrm{f}}(t), \omega_{\mathrm{f}}(t)), \tag{30}$$

where T_{f} and ω_{f} are the torque and speed of the engine, respectively; here, T_{f} is determined by

$$T_{\mathrm{f}}(t) + \kappa T_{\mathrm{m}}(t) = T_{\mathrm{dem}}(t), \tag{31}$$

wherein κ is the gear ratio between the engine and motor. It should be noted that the torque demand T_{dem} in the optimal control problem can be obtained by driver actions. Generally, T_{dem} is a function of the pedal position and the maximum torque, which is determined by a lookup table with the initial speed. In previous publications [17–19], several approaches to torque demand prediction have been proposed; in this study, we use the method proposed in [17].

For online implementation, approximate closed-form expressions are used instead of (29), e.g., polynomial expressions [20, 21]. In this paper, for analytical expressions, the motor power $P_{\mathrm{m}}(t)$ and the engine fuel rate $m_{\mathrm{f}}(t)$ are approximated as an affine piece-wise second order function of the output torque and speed, as

$$\begin{aligned} P_{\mathrm{m}}(t) &= \sum_{i=0}^{2}\sum_{j=0}^{2} a_{i,j} T_{\mathrm{m}}(t)^i \omega_{\mathrm{m}}(t)^j, \\ m_{\mathrm{f}}(t) &= \sum_{i=0}^{2}\sum_{j=0}^{2} b_{i,j} T_{\mathrm{f}}(t)^i \omega_{\mathrm{f}}(t)^j. \end{aligned} \tag{32}$$

There are also two kinematic equality constraints between velocities

$$\begin{aligned} \omega_{\mathrm{m}}(t) &= \kappa \frac{I_{\mathrm{g}}}{r_{\mathrm{w}}} v(t), \\ \omega_{\mathrm{f}}(t) &= \frac{I_{\mathrm{g}}}{r_{\mathrm{w}}} v(t), \end{aligned} \tag{33}$$

where $v(t)$ is the vehicle speed with the following dynamics:

$$\frac{\mathrm{d}}{\mathrm{d}t}v(t) = \frac{T_{\mathrm{dem}}(t)I_{\mathrm{g}}}{Mr_{\mathrm{w}}} - \frac{1}{2M}\rho c_{\mathrm{d}}A_{\mathrm{f}}v^2(t) - c_{\mathrm{r}}g, \tag{34}$$

where $I_{\mathrm{g}} \in \{16.41, 9.65, 6.32, 4.75, 3.63, 2.78\}$ is the transmission gear ratio; M is the equivalent mass; $\frac{1}{2}\rho c_{\mathrm{d}}A_{\mathrm{f}}v^2(t)$ is the aerodynamic drag resistance; ρ, c_{d}, and A_{f} are air density, air resistance coefficient, and face area, respectively; $c_{\mathrm{r}}g$ represents the acceleration caused by the rolling resistance and the gradient resistance, which is determined by the road slope θ, as

$$c_{\mathrm{r}} = f\cos(\theta) + \sin(\theta). \tag{35}$$

Here, f is the coefficient of rolling resistance. Given the transmission gear ratio I_{g}, which is obtained by a gear shift map, assuming the torque demand constraint in (31) holds, the vehicle velocity profile along the prediction horizon can be uniquely determined by the initial velocity and driver torque profile [16]. In summary, the optimization problem can be expressed as minimizing (27), subject to system dynamics (28) and (34), with control variables defined as $u = \gamma(t) = T_{\mathrm{m}}(t)/T_{\mathrm{dem}}(t)$ and states defined as $x = \mathrm{SoC}(t)$, under the given gearshift schedules and torque requirements.

Control inputs of the torque-split problem $T_{\mathrm{f}}(t)$ and $T_{\mathrm{m}}(t)$ are bounded by limitations $T_{\{\mathrm{f,m}\},\max}$ and $T_{\{\mathrm{f,m}\},\min}$ that typically vary with speed, since both engines and electric motors have torque peak curves that are functions of their respective rotational speeds. In addition to these limits, some other physical constraints must be enforced, as follows:

$$\begin{aligned}
&\gamma(t) \leqslant 1,\\
&\mathrm{SoC}_{\min} \leqslant \mathrm{SoC}(t) \leqslant \mathrm{SoC}_{\max},\\
&T_{\mathrm{m,min}} \leqslant T_{\mathrm{m}}(t) \leqslant T_{\mathrm{m,max}},\\
&0 \leqslant T_{\mathrm{f}}(t) \leqslant T_{\mathrm{f,max}},\\
&0 \leqslant \omega_{\mathrm{m}}(t) \leqslant \omega_{\mathrm{m,max}},\\
&0 \leqslant \omega_{\mathrm{f}}(t) \leqslant \omega_{\mathrm{f,max}}.
\end{aligned} \tag{36}$$

It should be noted that when in real-time applications, modelling parameters such as current SoC, battery open-circuit voltage V_{oc}, internal resistance R_{bat}, reference $\mathrm{SoC_r}$ and total mass M should be tuned before optimizing the problem (27) and (28). Based on the previous equations, the torque-split optimization problem can be represented and solved by the proposed GPM-MPC. In this study, we utilize the commercial optimization software, SNOPT [22], in which an SQP algorithm is used.

4.2 Evaluation of the GPM-MPC for torque-split optimization

This section presents simulation results that demonstrate the validity of the proposed GPM-MPC method. The driving cycles used in this section are the New European Driving Cycle (NEDC) and the Urban Dynamometer Driving Schedule (UDDS).

4.2.1 *Computational accuracy of the two methods*

Before showing the simulation results over standard driving cycles, we first discuss a driving scenario that includes acceleration from 0 to 16 m/s and a constant cruise, which is useful for choosing the number of discrete points. The velocity, gear shift schedule, and torque demand of the driving scenario are shown in Figure 8. To observe the performance of the proposed method, we compare the errors of the GPM-MPC and Euler-MPC. The sampling time is determined by the discrete point number N, the prediction horizon is T as $\Delta t = T/N$, and we set $T = 5$ s.

Neglecting the computational capacity of the hardware, for the Euler method, it is apparent that the larger the discrete point number is, the more accurate the solution that can be obtained. Thus, in this section, because of the infeasibility of the analytical solution, an approximate optimal result is chosen as

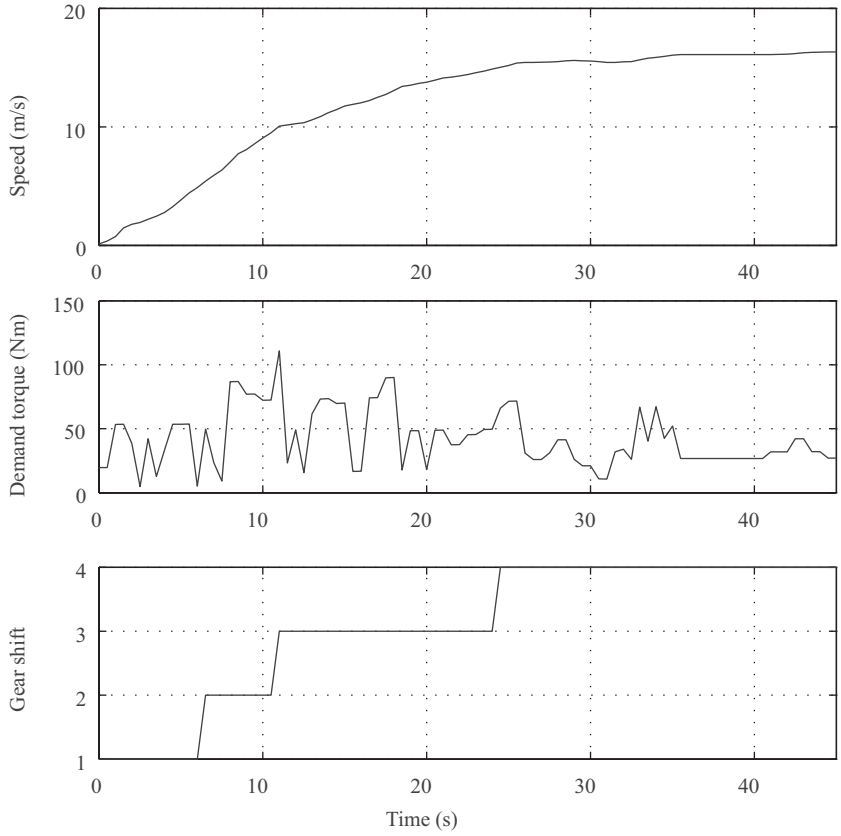

Figure 8 A driving scenario for evaluating the computational efficiency of the two methods.

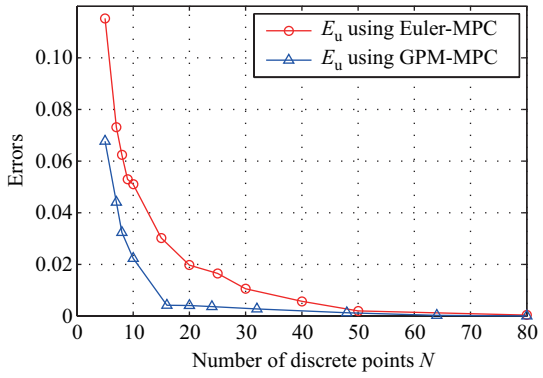

Figure 9 (Color online) Errors of the control variables and states using the GPM and Euler method in a driving scenario.

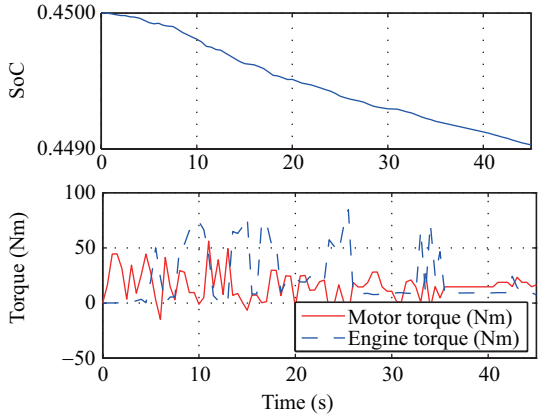

Figure 10 (Color online) State-of-charge and torque trajectories using the GPM-MPC ($N_g = 16$).

the benchmark with a larger number of discrete points, $N_{\text{exact}} = 200$, using the Euler method. Then, the error is defined as

$$E_u = \frac{\left|J_{\{g,e\}} - J_{\text{exact}}\right|}{J_{\text{exact}}}, \tag{37}$$

where J_g and J_e are the objective functions obtained by the GPM-MPC and Euler-MPC, respectively and J_{exact} is the exact objective value. Figures 9 and 10 show the results of the driving scenario. It can be seen that with the same number of discrete points, the computational accuracy of the GPM is much

Table 4 Fuel consumption using different methods under NEDC

Controller	Initial SoC	Final SoC	Fuel (g)
GPM-MPC method	0.6	0.466	–
	0.466	0.463	–
	0.463	0.463	284
Euler-MPC method	0.6	0.464	–
	0.464	0.458	–
	0.458	0.458	284

Table 5 Fuel consumption using different methods under UDDS

Controller	Initial SoC	Final SoC	Fuel (g)
GPM-MPC method	0.6	0.446	–
	0.446	0.446	396
Euler-MPC method	0.6	0.443	–
	0.443	0.443	419

higher than that of the Euler method, and the errors in the GPM decrease rapidly to a feasible level as the number of collocation points increases.

4.2.2 *Performance comparison over standard driving cycles*

A simulation was conducted by comparing the energy consumption of the proposed method (GPM-MPC) with that using the Euler method (Euler-MPC). The parameters are the same as in the above simulation, as are the NEDC and UDDS. The sampling time and discrete point number were chosen as $\Delta t = 0.5$ s and $N_{\mathrm{g}} = N_{\mathrm{e}} = 10$, with prediction horizon $T = 5$ s.

Tables 4 and 5 compare the fuel consumption values with equal initial and final SoCs for different controllers run on the same model of HEV. For fair comparison, the effect of different initial and final SoCs on fuel consumption should be considered. When the final SoC is lower than the initial SoC, fuel consumption will be lower. In order to remove this effect, we ran the simulations over the same cycle multiple times to reach a charge balance. Figures 11 and 12 show the results over the standard driving cycles with the same initial and terminal SoCs.

Figures 11 and 12 illustrate that, although the formulation of the optimization problem is the same, different discrete methods lead to different control results. Neglecting the computing accuracy of the converted NLP, different discrete methods and points may influence the approximate optimal control values, which will further influence the control effect. For the same number of discrete points, the energy efficiency obtained using the GPM-MPC is better than that using the Euler-MPC by 3%–4% on average, particularly for UDDS.

To evaluate the performance improvement of computational efficiency resulting from the proposed method, a simulation was performed aiming at comparing the computational time of the two methods while maintaining the same accuracy. The simulations were run on an Intel(R) Core(TM) i7-4790 CPU (3.60 GHz), and an estimate of CPU computational time was obtained using the CPU command in MATLAB. The computational times needed to solve the torque-split problem are shown in Figure 13. From these results, we can see that with the same energy efficiency (average error is 0.02 in Figure 9, and thus, the discrete point number is $N_{\mathrm{g}} = 10$ and $N_{\mathrm{e}} = 20$, respectively), the proposed method shows better potential for real-world implementation.

5 Conclusion

In this paper, a new formulation of model predictive control for continuous-time nonlinear systems is introduced; this method allows for the use of real-time optimization techniques. By combining the GPM and MPC, the finite-horizon optimal control problem is converted to an NLP problem. The main

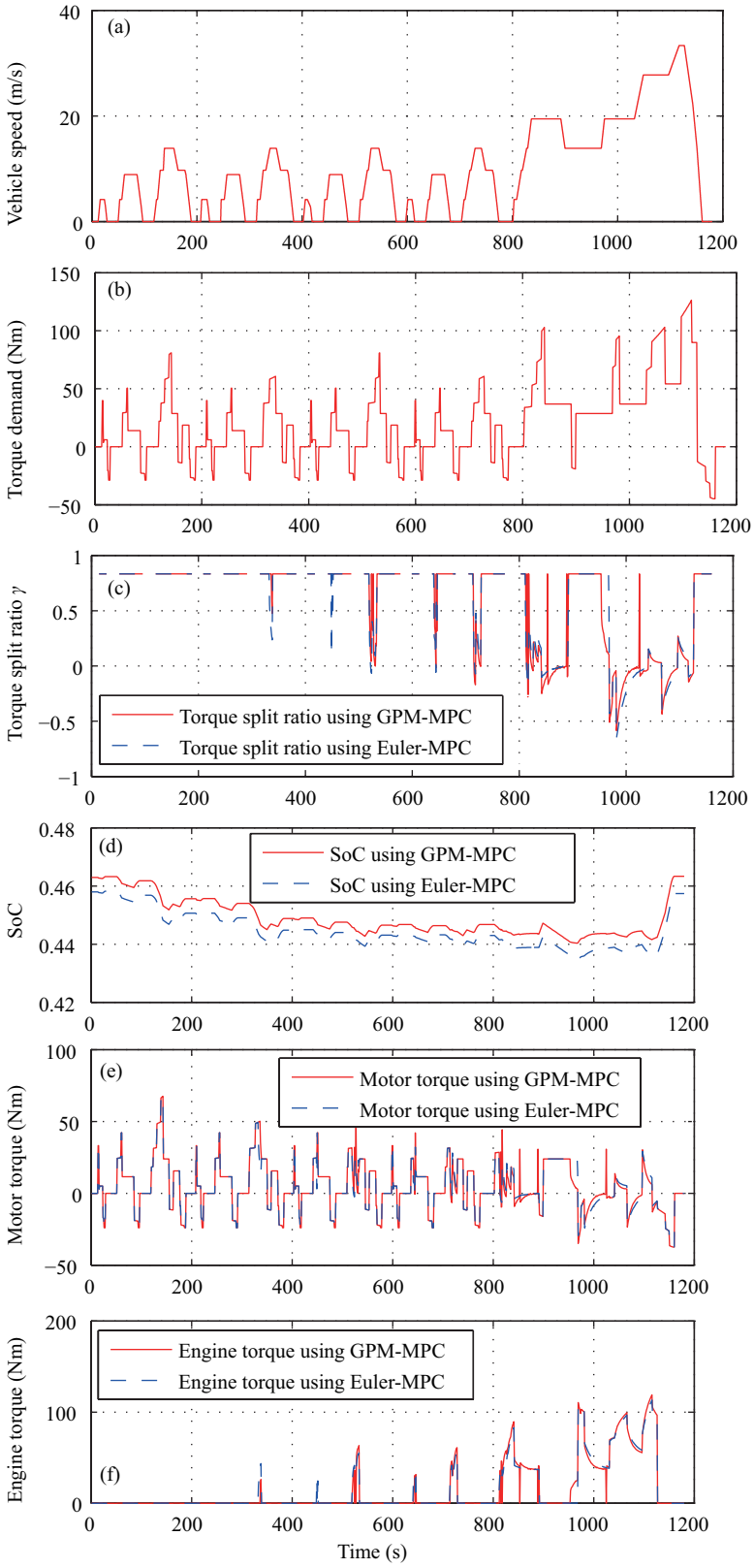

Figure 11 (Color online) Trajectories of simulation results using the GPM-MPC and Euler-MPC in NEDC. From top to bottom, the figures show the trajectories of (a) vehicle speed, (b) torque demand, (c) torque-split ratio, (d) state-of-charge, (e) motor torque, and (f) engine torque, respectively.

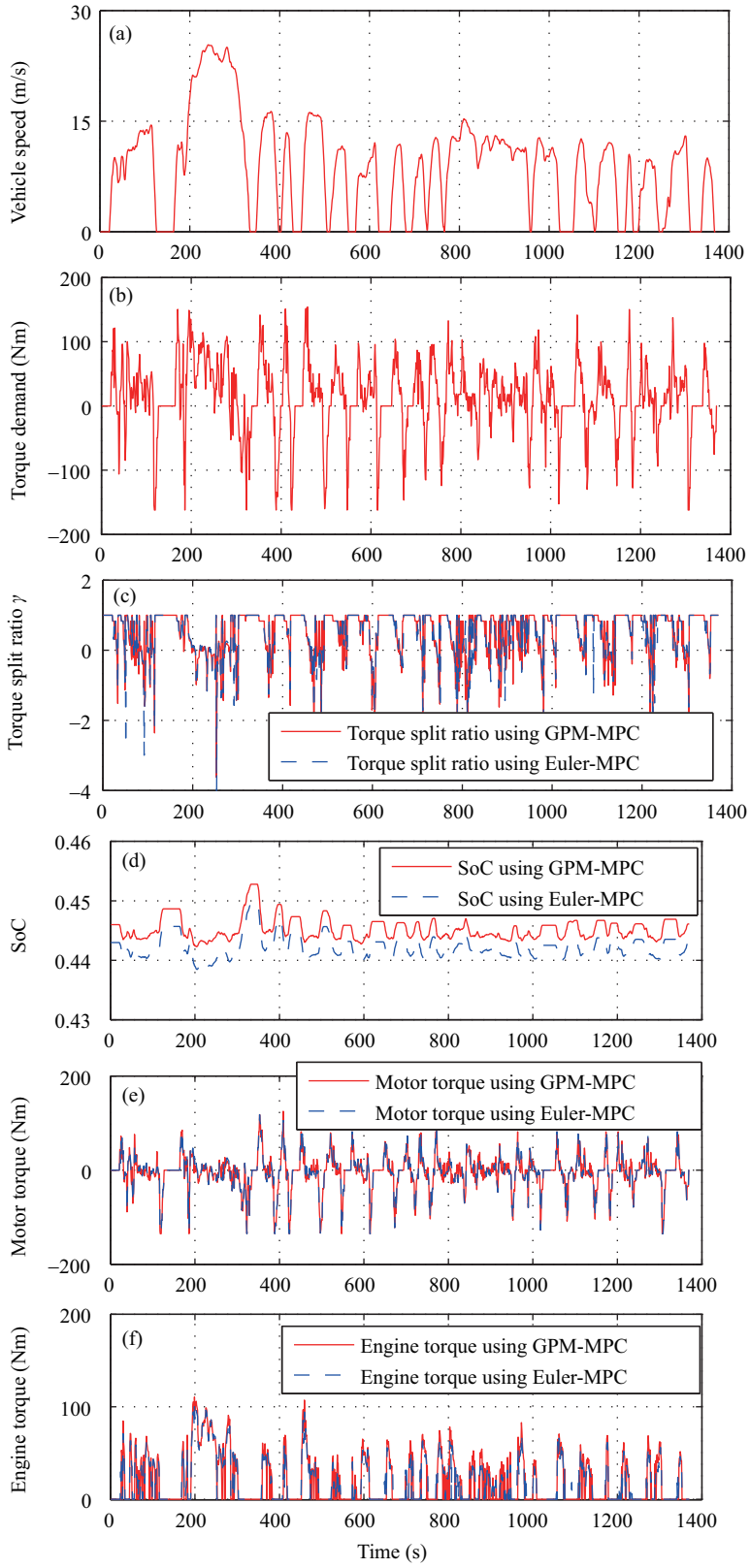

Figure 12 (Color online) Trajectories of simulation results using the GPM-MPC and Euler-MPC in UDDS. From top to bottom, the figures show the trajectories of (a) vehicle speed, (b) torque demand, (c) torque split ratio, (d) state-of-charge, (e) motor torque, and (f) engine torque, respectively.

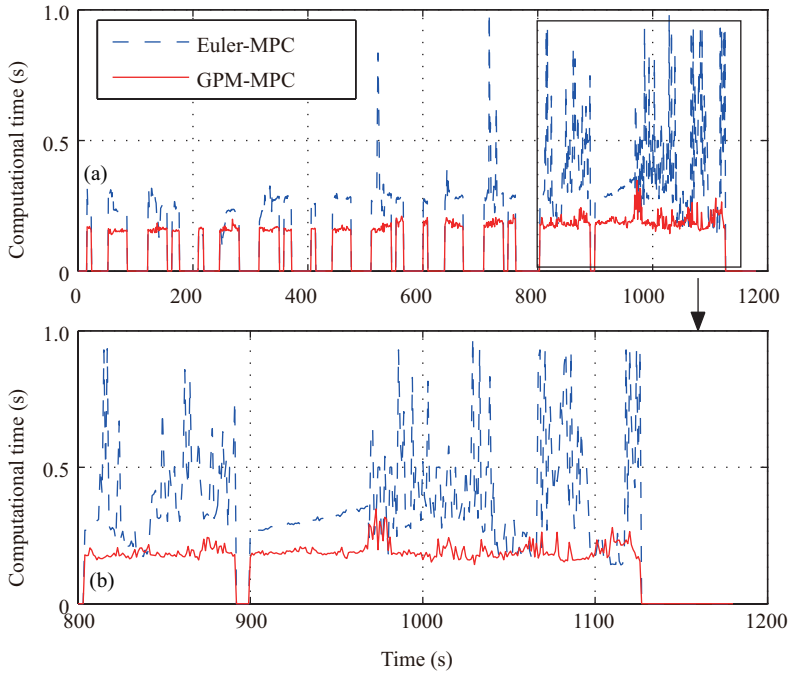

Figure 13 (Color online) Computational efficiency of the proposed method (NEDC). The second figure shows the results in the time horizon $t \in [800, 1180]$ s.

advantage of the proposed method is that it is less computationally demanding for online implementation compared with traditional discretization approaches, such as the Euler method.

To observe the effectiveness of the proposed method, several numerical examples are discussed in this paper, from which we conclude that the computational accuracy of the proposed method is superior to that of the Euler method. Using this solution, the optimal control of torque-split in HEVs based on known torque demand has been addressed. The novel algorithm calculates the optimal torque-split strategy by converting the problem to an NLP problem, and then solving the resulting NLP problem using the SQP algorithm, all with a computational effort of less than half that required for the Euler method, while ensuring the same accuracy.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61520106008, 61522307, 61374046) and Graduate Innovation Fund of Jilin University (Grant No. 2016188).

Conflict of interest The authors declare that they have no conflict of interest.

References

1 Grancharova A, Johansen T A. Explicit nonlinear model predictive control: theory and applications. In: Theory and Applications. Berlin: Springer, 2012. 429: 39–69
2 Kirches C. Fast Numerical Methods for Mixed-Integer Nonlinear Model-Predictive Control. Wiesbaden: Vieweg+ Teubner Verlag/Springer Fachmedien Wiesbaden, 2011
3 DeHaan D, Guay M. A real-time framework for model-predictive control of continuous-time nonlinear systems. IEEE Trans Autom Control, 2007, 52: 2047–2057
4 Betts J T. Survey of numerical methods for trajectory optimization. J Guid Control Dyna, 1998, 21: 193–207
5 Benson D A, Huntington G T, Thorvaldsen T P, et al. Direct trajectory optimization and costate estimation via an orthogonal collocation method. J Guid Control Dyna, 2006, 29: 1435–1440
6 Garg D, Patterson M A, Hager W W, et al. An overview of three pseudospectral methods for the numerical solution of optimal control problems. Adv Astronaut Sci, 2009, 135: 475–487
7 Benson D. A Gauss pseudospectral transcription for optimal control. Dissertation for Ph.D. Degree. Cambridge: Massachusetts Institute of Technology, 2005
8 Huntington G T. Advancement and analysis of a Gauss pseudospectral transcription for optimal control problems.

Dissertation for Ph.D. Degree. Cambridge: Massachusetts Institute of Technology, 2007
9 Xu S, Li S E, Zhang X, et al. Fuel-optimal cruising strategy for road vehicles with step-gear mechanical transmission. IEEE Trans Intell Transport Syst, 2015, 16: 3496–3507
10 Li S E, Xu S, Huang X, et al. Eco-departure of connected vehicles with V2X communication at signalized intersections. IEEE Trans Veh Tech, 2015, 64: 5439–5449
11 Burden R L, Faires J D, Reynolds A C. Numerical Analysis. Boston: Wadsworth Publishing Company, 1978
12 Sezer V, Gokasan M, Bogosyan S. A novel ECMS and combined cost map approach for high-efficiency series hybrid electric vehicles. IEEE Trans Veh Tech, 2011, 60: 3557–3570
13 Panday A, Bansal H O. A review of optimal energy management strategies for hybrid electric vehicle. Int J Veh Tech, 2014
14 Ambuhl D, Guzzella L. Predictive reference signal generator for hybrid electric vehicles. IEEE Trans Veh Tech, 2009, 58: 4730–4740
15 Adhikari S. Real-time power management of parallel full hybrid electric vehicles. Dissertation for Ph.D. Degree. Melbourne: The University of Melbourne. 2010
16 Borhan H, Vahidi A, Phillips A M, et al. MPC-based energy management of a power-split hybrid electric vehicle. IEEE Trans Control Syst Tech, 2012, 20: 593–603
17 Guo L L, Gao B Z, Chen H. On-line shift schedule optimization of 2-speed electric vehicle using moving horizon strategy. IEEE/ASME Trans Mech, 2016, 21: 2858–2869
18 Borhan H A, Vahidi A, Phillips A M, et al. Predictive energy management of a power-split hybrid electric vehicle. In: Proceedings of IEEE American Control Conference, St. Louis, 2009. 3970–3976
19 Zeng X, Huang K, Meng F. Model predictive control for parallel hybrid electric vehicles with potential real-time capability. J Autom Safe Energy, 2012, 3: 165–172
20 Sciarretta A, de Nunzio G, Ojeda L L. Optimal ecodriving control: energy-efficient driving of road vehicles as an optimal control problem. IEEE Control Syst, 2015, 35: 71–90
21 Dib W, Chasse A, Moulin P, et al. Optimal energy management for an electric vehicle in eco-driving applications. Control Eng Pract, 2014, 29: 299–307
22 Gill P E, Murray W, Saunders M A. SNOPT: an SQP algorithm for large-scale constrained optimization. SIAM Rev, 2005, 47: 99–131