Mechanical Systems and Signal Processing 87 (2017) 138–152

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

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

A model predictive speed tracking control approach for autonomous ground vehicles

Min Zhu, Huiyan Chen *, Guangming Xiong

School of Mechanical Engineering, Beijing Institute of Technology, Beijing, China

ARTICLE INFO

Article history:
Received 9 September 2015
Received in revised form
29 February 2016
Accepted 1 March 2016
Available online 30 March 2016

Keywords:
Speed tracking
Speed control
Autonomous ground vehicles
Model predictive control
Real-time optimization

ABSTRACT

This paper presents a novel speed tracking control approach based on a model predictive control (MPC) framework for autonomous ground vehicles. A switching algorithm without calibration is proposed to determine the drive or brake control. Combined with a simple inverse longitudinal vehicle model and adaptive regulation of MPC, this algorithm can make use of the engine brake torque for various driving conditions and avoid high frequency oscillations automatically. A simplified quadratic program (QP) solving algorithm is used to reduce the computational time, and the approach has been applied in a 16-bit microcontroller. The performance of the proposed approach is evaluated via simulations and vehicle tests, which were carried out in a range of speed-profile tracking tasks. With a well-designed system structure, high-precision speed control is achieved. The system can robustly model uncertainty and external disturbances, and yields a faster response with less overshoot than a PI controller.

1. Introduction

Autonomous ground vehicles, as an important part of intelligent transportation system (ITS), are attracting more attention than ever before. Their control system usually consists of three modules: environment perception, planning and decision-making, and vehicle control [1–5]. The environment perception module obtains information on surroundings by external sensors, such as lasers, cameras and radar, and then fuses the information by building environment maps to determine drivable surfaces. The planning and decision-making module gathers and handles task information, and combines it with vehicle states and drivable surfaces information to determine the desired path and the speed profile. The vehicle control module coordinates the engine, brakes and steering to track the desired path and speed.

For autonomous ground vehicles, the desired speed is determined by a variety of factors. For instance, the robot Stanley, which won the 2005 DARPA Grand Challenge, uses path planner, health monitor, and speed recommender to set the desired vehicle speed [2]. The speed tracking control system needs to track the desired speed precisely, especially when autonomous ground vehicles are conducting complex tasks, such as autonomous overtaking. This structure can simplify the control system of autonomous ground vehicles, so the planning and decision-making module can focus on determining the desired path and the speed profile and make more sophisticated decisions. Other longitudinal vehicle control methods, such as adaptive cruise control (ACC), stop and go (SG), and full speed range adaptive cruise control (FSRA), always work with a human driver and just follow the movements' trend of the preceding vehicle [6–8].

* Corresponding author. Tel.: +86 10 68912526.
E-mail address: chen_h_y@263.net (H. Chen).

http://dx.doi.org/10.1016/j.ymssp.2016.03.003
0888-3270/

Nomenclature

a, a_{des}	the actual and desired vehicle accelerations
f_b	the function between a and p_b
f_{tr}, $f_{tr,0}$	the torque ratio and the stall torque ratio
F_{xdes}	the desired longitudinal force
H, G	the factors for QP problem
H_p, H_c	the prediction and control horizons
i_g, i_o	the ratio of transmission and the ratio from transmission output shaft to wheel
K, τ	the system gain and time constant
k	the current sampling time
k_b	the proportional coefficient of the applied brake torque to the brake master cylinder pressure
k_c	the coefficient of p_{bdes} to $-a_{des}$ ($a_{des}<0$)
k_f, k_r	the proportional coefficient in a single side of the front/rear axle
k_P, k_I	the gains for the PI controller
k_α, k_β	the proportional coefficients of the actuator control inputs to the speed error metric
m	the vehicle mass include sprung mass and unsprung mass of both axles
n_e	the engine speed
n_t, n_o	the transmission input shaft speed and output shaft speed
p_b, p_{bdes}	the actual and desired brake master cylinder pressures
Q, R, S	the weighting matrices of the system output, control increment and control input
r_w	the effective radius of wheel
S_{tc}, $S_{tc,1}$	the speed ratio of torque converter and a given speed ratio when $f_{tr}=1$
T	the sampling period
T_{bdes}	the desired brake torque
T_{cdes}	the desired torque at the output side of torque converter
T_{edes}	the desired engine output torque
T_{emax}	the maximum engine torque
u	the control input
u_{min}, u_{max}	the acceleration limits
u_{PI}	the speed error metric
v, v_{ref}	the actual and desired vehicle speeds
α_{th}, α_{thdes}	the throttle control input and the desired throttle angle
β_{th}	the normalized brake control input
η_T	the powertrain efficiency
Δu^*	the optimal input increment
$\Delta u_{min,acc}$, $\Delta u_{max,acc}$	the acceleration increment limits in the drive mode
$\Delta u_{min,dec}$, $\Delta u_{max,dec}$	the acceleration increment limits in the brake mode
Δu_{min}, Δu_{max}	the limits of the final acceleration increments

Subscripts

qp	the Matlab QP solver quadprog was used
simp	the simplified QP solving algorithm was used
lpd	α_{th} is obtained from the look-up table of inverse engine map
elpd	α_{th} is obtained by a simplified engine mode

Some approaches have been proposed to enhance the speed tracking accuracy. Even if the gains of conventional PI/PID controller are well tuned for some operating regions, it is likely to demonstrate inferior performance in other conditions due to the severe nonlinearities present in the system. As a result, overshoot cannot be avoided, and the response is sluggish in most driving conditions [9,10]. Yanakiev et al. [9] introduced a signed quadratic (Q) term into the PID and adaptive PI controllers and used the proposed PIQD and adaptive PIQ controllers to reduce speed overshoot. The authors suggested that the response time becomes longer with reduced control gains. Hunt et al. [10,11] used a generalized gain scheduling approach to design a high precision speed controller. Proper selection of the observer polynomials is significant for making the controller insensitive to the measurement noise and un-modeled high frequency dynamics. Anderson proposed a Speed-based Acceleration Maps (SpAM)+PI controller to track the speed of an autonomous ground vehicle [12]. Although this configuration allows for a faster response with less overshoot than a PI controller, Anderson did not recommend implementing the approach because the generation of speed maps is both time consuming and sensitive to less than perfect data. Moreover, the throttle will oscillate causing a jerky ride when tracking a constant speed. Wang et al. [13] developed a nonparametric controller with an internal model control (IMC) structure for the longitudinal speed tracking control. The comparisons with SpAM+PI indicate that this system can track speeds in a wider range and yield an acceptable precision, but it is difficult to obtain an accurate nonparametric dynamical model for all the inputs. Fuzzy logic controller does not require a detailed model of the system, but considerable vehicle tests and parameter calibrations are needed to create a rule library that is large enough for use [14–17]. Kim et al. [1] developed a time-varying parameter adaptive vehicle speed controller that does not rely on a relatively accurate vehicle model, but its adaptation gains need to be tuned carefully to avoid high frequency oscillations. Murayama and Sakai et al. [18,19] applied a nonlinear model predictive control (NMPC) scheme for a torque demand control for speed tracking in vehicles with a variable valve lift engine, which is implemented by controlling the throttle angle, variable valve lift, ignition timing, and fuel injection directly. However, those control variables are not available for most researchers.

Like most other longitudinal vehicle control schemes, the proposed speed tracking controller has an upper level controller and a lower level controller. The upper level controller calculates the desired acceleration that “smoothly” and “quickly” track the desired speed profile. The lower level controller controls the engine and brakes to reach the desired acceleration [6,7,20]. The difference is that the proposed lower level controller does not need to track the desired

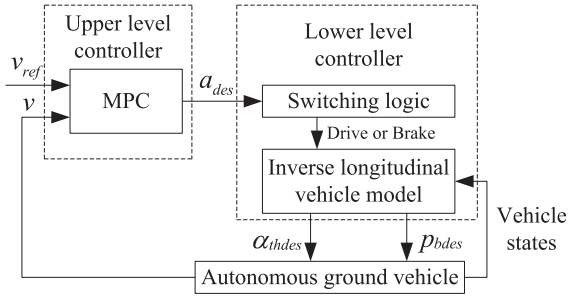

Fig. 1. The architecture of the speed tracking controller.

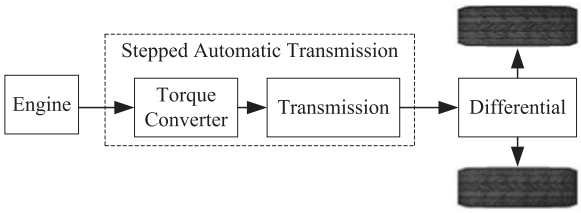

Fig. 2. Schematic diagram of a FWD or RWD vehicle's powertrain system with stepped automatic transmission.

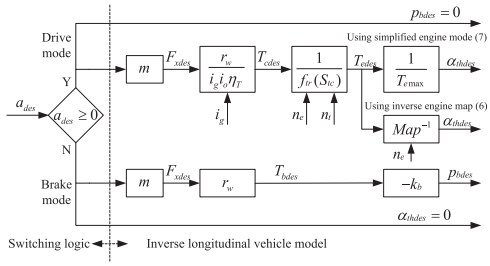

Fig. 3. The block diagram of the lower level controller.

acceleration command accurately, and the error is adjusted by using model predictive control (MPC) in the upper level controller. As such, the lower level controller only needs a simple inverse longitudinal vehicle model to reduce the effects of model error and the neglected dynamics in the linear vehicle model, especially when the gear ratio of stepped transmission changes.

This paper focuses on the engineering aspects and experimental evaluation of a test vehicle. We use our engineering experience to simplify the control system. A simple but practical method is used to calculate the desired throttle angle rather than an engine map, and only three vehicle states and four vehicle parameters that can be easily obtained in modern vehicles are used in the inverse longitudinal vehicle model. The feasibility of these simplifications is verified through simulations and vehicle tests, showing that the speed tracking control system can be quickly applied to other vehicles. A switching logic algorithm without calibration is proposed to determine the drive or brake control. Combined with the inverse longitudinal vehicle model and adaptive regulation of MPC, this algorithm can make use of the engine brake torque for various driving conditions and avoid high frequency oscillations automatically. Finally, the proposed controller can provide a good performance in a test vehicle, which is implemented in a 16-bit microcontroller.

The organization of this paper is as follows. Section 2 shows the design of the controller, a switching logic and a simple inverse longitudinal vehicle model for control. Then based on the framework of MPC, a predictive optimization problem is formulated and simplified for microcontroller. In Section 3, the performance of the controller is verified by simulations. Section 4 presents the test results for a range of speed-profile tracking tasks. Finally, the main conclusions are given.

2. Controller design

Fig. 1 shows the architecture of the speed tracking controller as described in the introduction. The desired acceleration command a_{des} is obtained by MPC in the upper level controller according to the desired vehicle speed v_{ref} and the actual vehicle speed v. Firstly, a switching logic is adopted in the lower level controller to determine the drive or brake control.

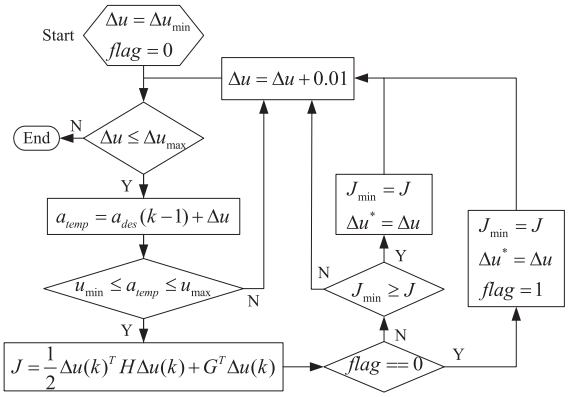

Fig. 4. QP problem solving algorithm for microcontroller.

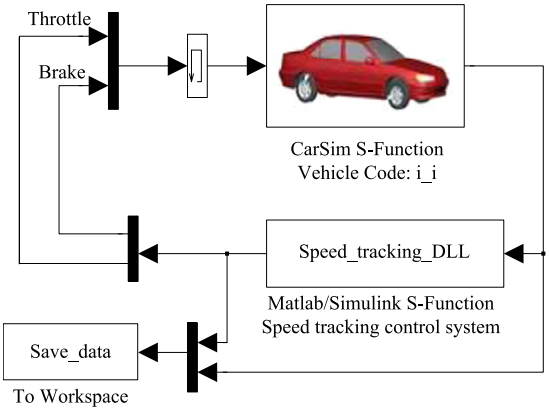

Fig. 5. Configuration of the simulation platform.

Table 1
Vehicle parameters of three classes of vehicles.

Parameter	A-Class (FWD, 5AT)	D-Class (FWD, 6AT)	E-Class (4WD, 7AT)
m	830 (kg)	1530 (kg)	1833 (kg)
r_w	0.292 (m)	0.33 (m)	0.359 (m)
i_o	4.1	4.1	2.65
η_T	0.9	0.9	0.9
$f_{tr,0}$	1.864	1.864	1.864
$S_{tc,1}$	0.88	0.88	0.88
T_{emax}	160 (N m)	320 (N m)	535 (N m)
k_f	150 (N m/MPa)	300 (N m/MPa)	400 (N m/MPa)
k_r	100 (N m/MPa)	150 (N m/MPa)	300 (N m/MPa)

Then actuator control inputs (the desired throttle angle α_{thdes} and the desired brake master cylinder pressure p_{bdes}) can be obtained by an inverse longitudinal vehicle model.

2.1. The lower level controller

Fig. 2 shows a schematic diagram of a front-wheel drive (FWD) or rear-wheel drive (RWD) vehicle's powertrain system with stepped automatic transmission (AT), which includes assemblies such as the engine, torque converter, transmission, differential gear and wheels. For a 4-wheel drive (4WD) system, a transfer case is also included and the same inverse longitudinal vehicle model can be used for control purposes.

Fig. 3 shows the block diagram of the lower level controller. Most literatures suggested that a switching line (obtained from data on zero-throttle acceleration of a vehicle) with boundary layers is used to determine the drive or brake control [21,22]. Instead, we compare a_{des} to zero, and if $a_{des} \ge 0$, the drive control is applied (drive mode), otherwise the brake

Table 2
Key parameters in cost function and constraints.

Parameter	Value	Parameter	Value (m/s^2)
Q	3	u_{min}	−5
R	5	u_{max}	3
S	1	$\Delta u_{min,acc}$	−0.5
H_p	20	$\Delta u_{max,acc}$	0.05
H_c	1	$\Delta u_{min,dec}$	−0.5
T	0.05 (s)	$\Delta u_{max,dec}$	1

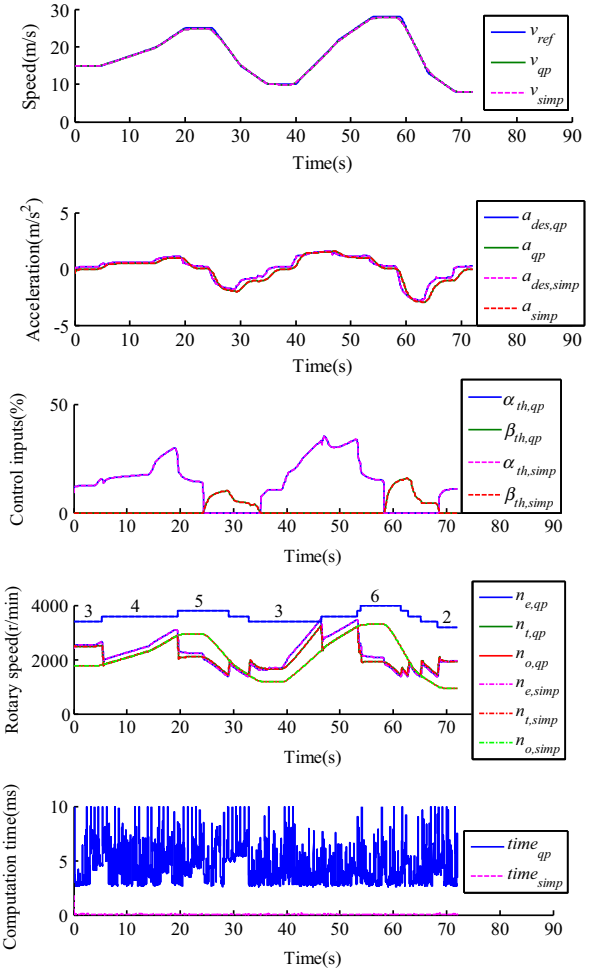

Fig. 6. Comparison of simulation results for the Matlab QP solver quadprog and the simplified QP solving algorithm (D-Class, Flat road, α_{th} is obtained from the look-up table of the inverse engine map (6)).

control is applied (brake mode). The switching logic can be formulated as follows:

$$\alpha_{thdes} = \begin{cases} \alpha_{thdes}, & a_{des} \geq 0 \\ 0, & a_{des} < 0 \end{cases}$$

$$p_{bdes} = \begin{cases} 0, & a_{des} \geq 0 \\ p_{bdes}, & a_{des} < 0 \end{cases} \tag{1}$$

In the drive mode, the inverse longitudinal vehicle model uses three vehicle states (engine speed n_e, transmission input shaft speed n_t, and ratio of transmission i_g) and four vehicle parameters (vehicle mass m, effective radius of wheel r_w, ratio from transmission output shaft to wheel i_o, and powertrain efficiency η_T) to calculate the throttle control input. The desired

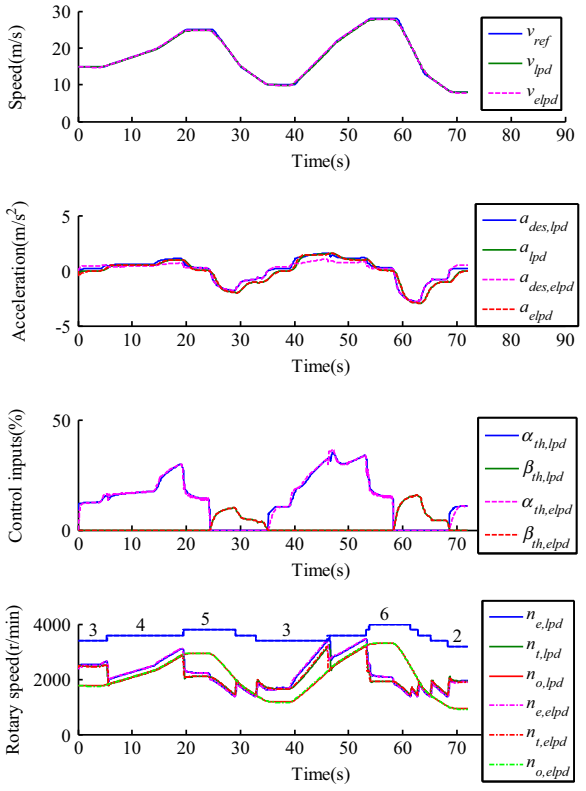

Fig. 7. Comparison of simulation results for different engine modes (D-Class, Flat road).

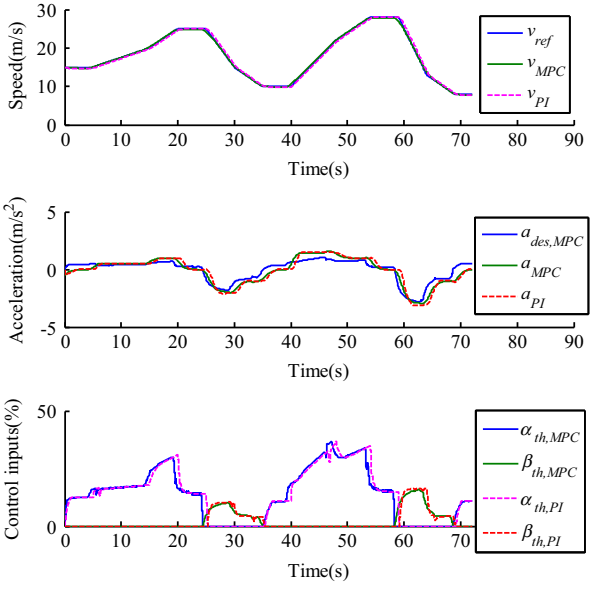

Fig. 8. Comparison of simulation results for MPC and PI controllers (D-Class, Flat road).

longitudinal force acting on wheels (F_{xdes}) is computed from the following equation:

$$F_{xdes} = ma_{des} \tag{2}$$

The desired torque at the output side of torque converter (T_{cdes}) is derived from:

$$T_{cdes} = \frac{F_{xdes} r_w}{i_g i_o \eta_T} \tag{3}$$

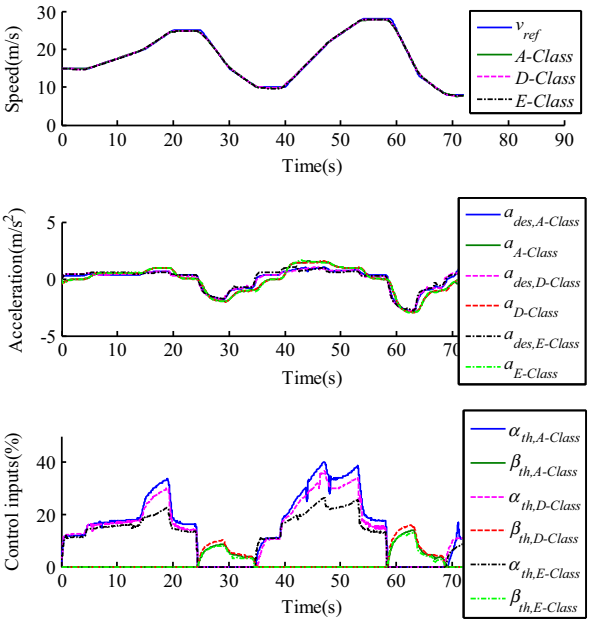

Fig. 9. Comparison of simulation results for three types of vehicles (Flat road).

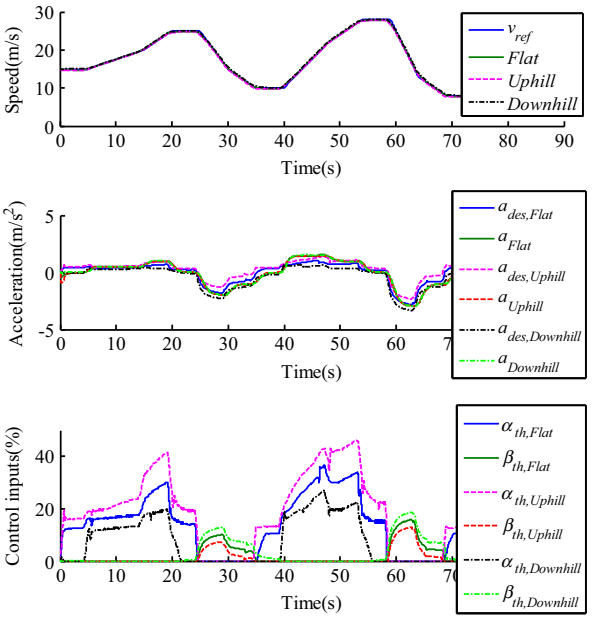

Fig. 10. Comparison of simulation results for different road grades (D-Class).

The desired engine output torque T_{edes} is derived from:

$$T_{edes} = \frac{T_{cdes}}{f_{tr}(S_{tc})} \tag{4}$$

where $f_{tr}(S_{tc})$ is the torque ratio of torque converter as a function of the speed ratio S_{tc}, and $S_{tc}=n_t/n_e$. Although the characteristics of torque converter change with the temperature of automatic transmission fluid (ATF) and the input speed of torque converter, we can use the following equation to simplify computation:

$$f_{tr}(S_{tc}) = \begin{cases} f_{tr,0} - \frac{f_{tr,0}-1}{S_{tc,1}} S_{tc}, & 0 \le S_{tc} < S_{tc,1} \\ 1, & S_{tc} \ge S_{tc,1} \end{cases} \tag{5}$$

Table 3
RMS errors for various simulation conditions.

	Flat (m/s)	Uphill (m/s)	Downhill (m/s)
A-Class, MPC, *lpd*	0.22	0.40	0.25
D-Class, MPC, *lpd*	0.20	0.37	0.24
E-Class, MPC, *lpd*	0.18	0.38	0.22
A-Class, MPC, *elpd*	0.23	0.30	0.29
D-Class, MPC, *elpd*	0.21	0.28	0.28
E-Class, MPC, *elpd*	0.22	0.30	0.24
A-Class, PI	0.45	0.52	0.41
D-Class, PI	0.45	0.49	0.41
E-Class, PI	0.36	0.38	0.34

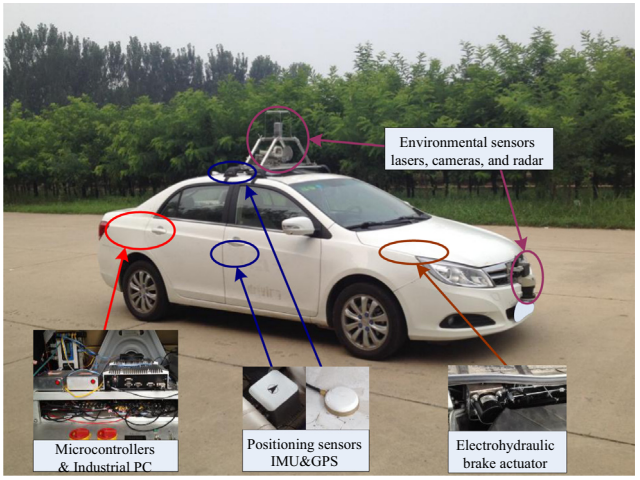

Fig. 11. Configuration of the test vehicle.

Fig. 12. A panorama of the test road.

where $f_{tr,0}$ is the stall torque ratio when $S_{tc}=0$, $S_{tc,1}$ is a given speed ratio when $f_{tr}=1$ for the first time. These two parameters can be obtained from the product manual of automatic transmission. We can obtain α_{thdes} by looking up the inverse engine map described by (6).

$$\alpha_{thdes} = Map^{-1}(T_{edes}, n_e) \tag{6}$$

Still, it is difficult to obtain a complete engine map for some academic researchers, and it cannot be transferred to other vehicles quickly. Thus, we use (7) instead of (6) to compute α_{thdes}, and the control effects are compared with each other in the simulation.

$$\alpha_{thdes} = \frac{T_{edes}}{T_{e\ max}} \cdot 100\% \tag{7}$$

where T_{emax} is the maximum engine torque and can be obtained from the engine's product manual. A simple brake system without boost and thermal effects is used in the simulation. The desired brake master cylinder pressure p_{bdes} can be calculated by the following equation:

$$p_{bdes} = -k_b T_{bdes} \tag{8}$$

where T_{bdes} is the desired brake torque, $T_{bdes}=F_{xdes}r_w$, k_b is the proportional coefficient of the applied brake torque to the

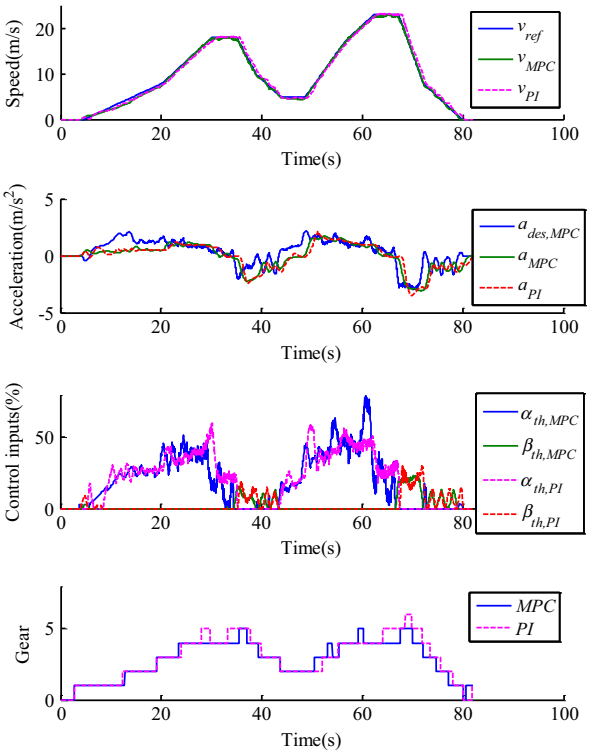

Fig. 13. Comparison of test results for MPC and PI controllers (*Ref1*).

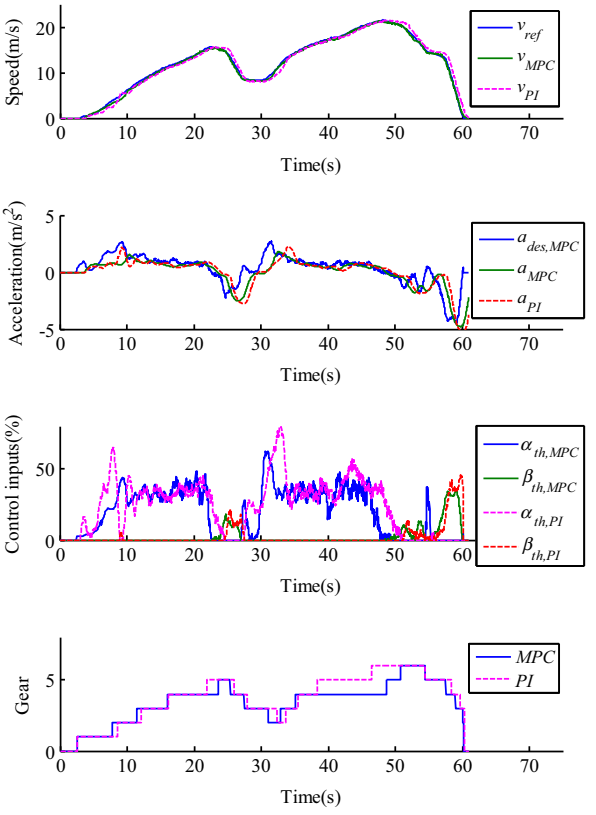

Fig. 14. Comparison of test results for MPC and PI controllers (*Ref2*).

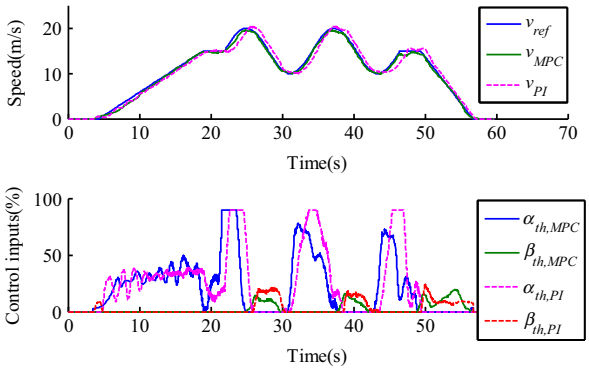

Fig. 15. Comparison of test results for MPC and PI controllers (*Ref3*).

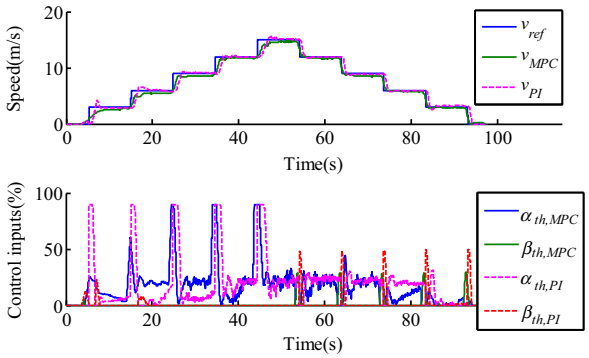

Fig. 16. Comparison of test results for MPC and PI controllers (*Ref4*).

brake master cylinder pressure, it can be computed from the following equation:

$$k_b = 2(k_f + k_r) \tag{9}$$

where k_f is the proportional coefficient of the applied brake torque between the wheel and the brake cylinder pressure in a single side of the front axle and k_r is the proportional coefficient in a single side of the rear axle. We use a solenoid valve to control the brake master cylinder pressure in the test vehicle. To reduce calibration work, we tested the vehicle and found $a=f_b(p_b)$, which simplifies the vehicle acceleration a as a function of the brake master cylinder pressure p_b. Then, we can use (10) to calculate p_{bdes} for a given a_{des}.

$$p_{bdes} = f_b^{-1}(a_{des}) = -k_c a_{des} \quad , a_{des} < 0 \tag{10}$$

where k_c is a coefficient obtained by the linear least-squares (LLS) method, and $k_c=0.8\ \text{MPa}\,(\text{m/s}^2)^{-1}$ for the test vehicle.

2.2. Model predictive control

In many previous studies [23–25], a first order system is applied to model the lower level controller:

$$a = \frac{K}{\tau s + 1} a_{des} \tag{11}$$

where $K=1.0$ is the system gain, τ is the time constant, $\tau=0.2$ and 0.5 for the simulation vehicles and the test vehicle, respectively. Considering the relationship between speed and acceleration, a two-state space model for speed tracking control system can be formulated as:

$$\begin{aligned} &\dot{x} = \Phi x + \Pi u \\ &\Phi = \begin{bmatrix} 0 & 1 \\ 0 & -1/\tau \end{bmatrix}, \Pi = \begin{bmatrix} 0 \\ K/\tau \end{bmatrix} \\ &x = [v, a]^T, u = a_{des} \end{aligned} \tag{12}$$

where $x \in R^2$ is a vector of the system states, and $u \in R$ is the control input. The discrete-time expression (13) is computed

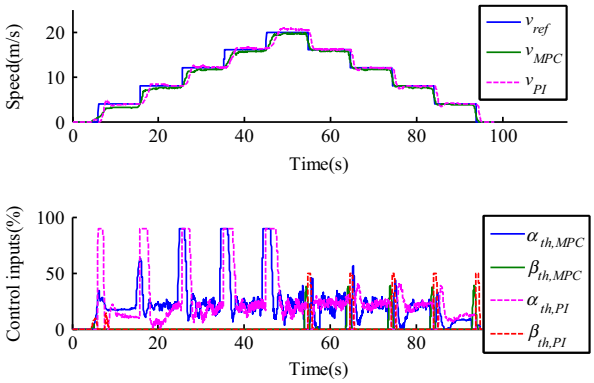

Fig. 17. Comparison of test results for MPC and PI controllers (*Ref5*).

Table 4
RMS errors for various reference speed profiles.

	Ref1 (m/s)	*Ref2* (m/s)	*Ref3* (m/s)	*Ref4* (m/s)	*Ref5* (m/s)
MPC, *elpd*	0.35	0.33	0.47	0.66	0.91
PI	0.66	0.75	1.02	0.93	1.29

using the forward Euler (FE) discretization method:

$$x(k+1) = Ax(k) + Bu(k)$$
$$A = \begin{bmatrix} 1 & T \\ 0 & 1 - T/\tau \end{bmatrix}, B = \begin{bmatrix} 0 \\ KT/\tau \end{bmatrix} \tag{13}$$

where k is the current sampling time, $k+1$ is the next sampling time, and T is sampling period. We use v as the system output, so the output equation is:

$$y(k) = Cx(k), C = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{14}$$

The system's objective is speed tracking accuracy. At the same time, by penalizing excessive acceleration or changes in acceleration, the cost function can be expressed as:

$$J(x(k), u(k-1), \Delta u(k)) = \sum_{i=1}^{Hp} ||y_p(k+i|k) - y_{ref}(k+i|k)||_Q^2 + \sum_{i=0}^{Hc-1} \|\Delta u(k+i)\|_R^2 + \sum_{i=0}^{Hc-1} \|u(k+i)\|_S^2 \tag{15}$$

where $k-1$ is the previous sampling time, H_p, H_c is the prediction and control horizon, respectively, $y_p(k+i|k)$ and $y_{ref}(k+i|k)$, with $i=1, \ldots, H_p$, are the prediction and reference for the output tracking variable, respectively, $(k+i|k)$ denotes the value predicted at time $k+i$ based on the information available at time k, $u(k+i)$ and $\Delta u(k+i)$, with $i=0, \ldots, H_c-1$, are the control input and control increment at time $k+i$, respectively, and Q, R and S refer to the weighting matrices of the system output, control increment and control input, respectively. The system constraints include the acceleration and its rate of change. The rate of change is different for the drive and brake modes.

$$u_{min} \le u(k+i) \le u_{max}, \quad i = 0, \cdots, H_c - 1 \tag{16}$$

$$\Delta u_{min,acc} \le \Delta u_{acc}(k+i) \le \Delta u_{max,acc}, i = 0, \cdots, H_c - 1$$
$$\Delta u_{min,dec} \le \Delta u_{dec}(k+i) \le \Delta u_{max,dec}, i = 0, \cdots, H_c - 1 \tag{17}$$

where u_{min} and u_{max} are the acceleration limits, $\Delta u_{min,acc}$ and $\Delta u_{max,acc}$ are the acceleration increment limits in the drive mode when the drive control is released and actualized, respectively, and $\Delta u_{min,dec}$ and $\Delta u_{max,dec}$ are the acceleration increment limits in the brake mode when the brake control is actualized and released, respectively. We use selection logic (19) to determine the limits of the final acceleration increments (18).

$$\Delta u_{min} \le \Delta u(k+i) \le \Delta u_{max}, i = 0, \cdots, H_c - 1 \tag{18}$$

$$\Delta u_{max} = \begin{cases} \Delta u_{max,acc} & , a_{des}(k-1) \ge 0 \\ min\{\Delta u_{max,dec}, \\ \Delta u_{max,acc} - a_{des}(k-1)\} & , a_{des}(k-1) < 0 \end{cases}$$

$$\Delta u_{min}=\begin{cases} max\{\Delta u_{min\ ,acc}, & ,a_{des}(k-1)>0 \\ \Delta u_{min\ ,dec}-a_{des}(k-1)\} & \\ \Delta u_{min\ ,dec} & ,a_{des}(k-1)\le 0 \end{cases} \tag{19}$$

where $a_{des}(k-1)$ is the desired acceleration command at the previous sampling time $k-1$. Logic (19) ensures a smooth transition from the drive mode to the brake mode or from the brake mode to the drive mode when a_{des} needs to pass through zero. Δu_{min} and Δu_{max} are finally adopted as the acceleration increment limits in each control cycle. The fundamental goal of MPC is to minimize the cost function (15) while satisfy the control constraints (16) and (18). We solve the following optimization problem at each time step.

$$\min_{\Delta u(k)} \quad J(x(k),u(k-1),\Delta u(k)) \tag{20}$$

Subject to: (a) Plant model – (13);
(b) Control constraints – (16) and (18).

The optimization problem (20) can be translated into a standard quadratic program (QP). To implement the speed tracking controller in a 16-bit microcontroller, we set $H_c=1$ to reduce the computational time. Then, the QP problem can be simplified as (21). The peak operation time is less than 30 ms, thus we can set the sampling period $T=50$ ms in the test vehicle.

$$\begin{aligned} \min_{\Delta u(k)} \quad & \frac{1}{2}\Delta u(k)^T H\Delta u(k)+G^T\Delta u(k) \\ s.t. \quad & \Delta u_{min}\le \Delta u(k)\le \Delta u_{max} \\ & u_{min}\le a_{des}(k-1)+\Delta u(k)\le u_{max} \end{aligned} \tag{21}$$

where $H\in R$ and $G\in R$ are factors for QP problem. The accuracy of a_{des} is set to 1% because the control accuracy of the drive and brake actuators is finite. For example, according to the control protocol provided by the engine manufacturer, the throttle control accuracy is 0.4%, and the control accuracy of brake solenoid valve is 0.5% in our design. The actuators cannot respond to the desired acceleration to a higher degree of accuracy. So we can use the algorithm as shown in Fig. 4 to solve QP problem (21), and the control performance is compared with the Matlab QP solver quadprog in the simulation.

Then, the optimal input increment Δu^* is obtained to compute the optimal control input:

$$u(k)=u(k-1)+\Delta u^* \tag{22}$$

When we combine the inverse longitudinal vehicle model and switching logic, we will find that there is no fixed relationship between a_{des} and a. For example, when a_{des} is equal to zero, T_{edes} is also equal to zero, the vehicle is coasting, and a varies with the driving condition. Therefore, the MPC algorithm in the upper level controller regulates a_{des} to obtain a real acceleration, which can reduce the speed tracking error. It is this design ensures the system performance and automatically makes use of the engine brake torque as a_{des} goes to 0, and also eliminates the impact of the rolling resistance, aerodynamic drag and road slope. Moreover, the proposed switching logic does not need to calibrate the switching threshold between the drive and brake, which can avoid a high frequency of oscillation automatically.

3. Simulation results

The configuration of the simulation platform is shown in Fig. 5. To move the control algorithm of the speed tracking control system directly to the microcontroller, we use standard C language to program the algorithm in a dynamic link library (DLL), and MATLAB/Simulink (The Mathworks, Natick, Massachusetts, USA) calls the DLL to control the vehicle dynamic model from the vehicle simulation software Carsim (The Mechanical Simulation Corporation, Ann Arbor, Michigan, USA, version: 8.02). To facilitate comparison with other publications and verify the robustness of the system, three different full vehicle dynamic models with default vehicle parameters from Carsim were chosen for simulation. The A-Class Hatchback is a FWD vehicle with a 75 kW engine and 5-speed AT, the D-Class Sedan is a FWD vehicle with a 150 kW engine and 6-speed AT, and the E-Class Sedan is a 4WD vehicle with a 250 kW engine and 7-speed AT. The simulation vehicle parameters for the inverse longitudinal vehicle model are listed in Table 1. We use n_t/n_o instead of i_g to achieve a smooth shifting process of transmission, where n_o is transmission output shaft speed.

To verify the tracking performance at various vehicle speeds, we have designed a reference desired speed profile consisting of different accelerations: -3 m/s^2, -2 m/s^2, -1 m/s^2, 0 m/s^2, 0.5 m/s^2, 1 m/s^2, and 1.5 m/s^2. This desired speed profile is conceived to evaluate the system over a wide range of operation. Key parameters in the cost function and constraints are listed in Table 2. These parameters are used for all the simulations and vehicle tests.

Firstly, the Matlab QP solver quadprog and simplified QP solving algorithm are compared with each other. Fig. 6 shows the results for the D-Class vehicle on a flat road. Where α_{th} is the throttle control input, $\alpha_{th}=\alpha_{thdes}$, which is obtained from the look-up table of the inverse engine map (6), β_{th} is the normalized brake control input, $\beta_{th}=p_{bdes}/(10\ \text{MPa})\times 100\%$, subscript *qp* denotes that the Matlab QP solver quadprog was used, and subscript *simp* denotes that the simplified QP solving algorithm was used. They have the same control effects, but the computational time is greatly reduced when the

simplified QP solving algorithm is used. And the same is true when α_{th} is obtained by the simplified engine mode (7), so we use the simplified QP solving algorithm in all the following simulations and vehicle tests.

We use *lpd* to show that α_{th} is obtained from the look-up table of the inverse engine map (6), *elpd* express α_{th} is obtained by the simplified engine mode (7), and compare the simulation results for the two engine modes in Fig. 7. As the engine models are different, the MPC controller uses different desired acceleration commands to obtain similar actual vehicle accelerations to ensure the speed tracking is accurate in the drive mode for both situations. So they have similar throttle control inputs, and the shifting processes of the transmission are approximately the same for both. Using the simplified engine mode (7) for control is proved to be feasible and effective. Thus, we use it for all the following simulations and vehicle tests.

A PI controller is designed for the sake of comparison, and the control law is:

$$u_{PI} = k_P(v_{ref} - v) + k_I \int_0^t (v_{ref} - v)dt \tag{23}$$

$$\alpha_{thdes} = \begin{cases} k_\alpha u_{PI}, & u_{PI} \geq 0 \\ 0, & u_{PI} < 0 \end{cases}$$
$$p_{bdes} = \begin{cases} 0, & u_{PI} \geq 0 \\ k_\beta u_{PI}, & u_{PI} < 0 \end{cases} \tag{24}$$

where k_P=0.4, k_I=0.001 are well tuned gains for the D-Class vehicle on a flat road, u_{PI} is a speed error metric, when the error metric is positive, the drive control is applied, otherwise the brake control is applied, and k_α=100% $(\mathrm{m/s})^{-1}$ as well as k_β=−5 MPa $(\mathrm{m/s})^{-1}$ are proportional coefficients of the actuator control inputs to the error metric. The simulation results for MPC and PI controllers are compared in Fig. 8. The root mean square (RMS) error values of the speed using MPC and PI controllers are 0.21 m/s and 0.45 m/s, respectively, showing that the proposed approach has a higher tracking accuracy. The MPC controller allows for a faster response with less overshoot than the PI controller, and the actuator control inputs of the PI controller response lags behind the MPC controller.

To verify the robustness of the proposed system's ability to model uncertainty, we have also simulated two other classes of vehicles on a flat road, and compared the simulation results in Fig. 9. The control performances are very similar for all vehicles with regulating a_{des} effectively. Although there are variations of vehicle parameters, the MPC controller select proper values of a_{des} to achieve a similar a value for each case, and good tracking performance is achieved regardless of vehicle type.

Constant road grades of 5° and −5° is considered as an external disturbance to the D-class vehicle to verify the robustness of the system. The simulation results are compared in Fig. 10. It can be seen that the MPC controller set $a_{des,Uphill} > a_{des,Flat} > a_{des,Downhill}$ to achieve similar a for different road grades, and control inputs $\alpha_{th,Uphill} > \alpha_{th,Flat} > \alpha_{th,Downhill}$, $\beta_{th,Uphill} < \beta_{th,Flat} < \beta_{th,Downhill}$ are obtained. The tracking performance is robust to the unpredictable external disturbance.

The RMS errors are compared in Table 3 for various simulation conditions. The proposed controller yields a higher tracking accuracy than the PI controller for all the simulations. There is always a time delay and sometimes overshoot with the PI algorithm, while the MPC controller shows a faster response with less overshoot. With adaptive regulation of the MPC controller, good tracking performance can be obtained even using the simplified engine mode (7) instead of the inverse engine map (6). The system can robustly model uncertainty and external disturbance.

4. Vehicle tests

To confirm the performance of the proposed speed tracking control system, vehicle tests have been conducted. The configuration of the test vehicle is shown in Fig. 11. The test vehicle is developed as an autonomous ground vehicle based on a passenger car. The carmaker authorizes us to use the throttle control protocol, but the brake system is still manual. To allow the brakes to be electronically controlled, we have designed an electrohydraulic brake system, which uses 8-bit pulse width modulator (PWM) to control a solenoid valve, and adjusts the brake master cylinder pressure with 0.5% accuracy. A 16-bit microcontroller is used for speed tracking control. A controller area network (CAN) bus connects all the electronic subsystems, so the speed tracking microcontroller can monitor the engine, transmission and electrohydraulic brake system states, and control the actuator control inputs directly.

To provide repeatable and identical speed profiles for controller comparison, environmental sensors such as lasers, cameras, and radar are not used, and the planning and decision-making module, which runs on an industrial personal computer (PC), send the desired speed profile to the speed tracking microcontroller directly, without being influenced by the environmental information. Positioning sensors such as inertial measurement units (IMUs) and GPS systems are mounted to estimate the location and speed of the vehicle relative to an external coordinate system. A Kalman filter is used to produce the vehicle acceleration signal from the vehicle speed as [1,24].

The tests were conducted on a flat road as shown in Fig. 12. Due to the space limitations and vehicle safety considerations, the maximum speed is limited to 25 m/s. We designed different reference desired speed profiles to compare the performance of MPC and PI controllers. *Ref1* denotes a reference speed profile consisting of different accelerations: −3 $\mathrm{m/s^2}$,

-2 m/s^2, -1 m/s^2, 0 m/s^2, 0.5 m/s^2, 1 m/s^2, and 1.5 m/s^2 as shown in Fig. 13. *Ref2* denotes a reference speed profile that is obtained from the manual driving data as shown in Fig. 14. *Ref3* denotes a reference speed profile which involves a sinusoidal speed profile as shown in Fig. 15. To evaluate the system performance more rigorously, we have also designed reference speed profiles *Ref4* and *Ref5*, which involve step changes of 3 m/s and 4 m/s every 10 s as shown in Figs. 16 and 17, respectively. Although this speed request is unreasonable, it can test the performance of the system quickly. RMS errors for various reference speed profiles are listed in Table 4. It can be seen that the proposed model predictive speed tracking control approach yields a higher tracking accuracy than the PI controller for all the vehicle tests. Similarly to the simulation results, the MPC controller shows a faster response with less overshoot than the PI controller. With the desired speed step changed, as shown in Figs. 16 and 17, there is always an overshoot with the PI controller, and the actuator control inputs response obviously lags behind those of the MPC controller.

5. Conclusions

A model predictive speed tracking control approach for autonomous ground vehicles is presented in this paper. With a well-designed system structure, the proposed approach achieves high-precision speed control for various simulations and vehicle tests. The speed tracking performance can robustly model uncertainty and external disturbances, and there is always a faster response with less overshoot than a PI controller. Some simplifications were verified and proven to be feasible and effective, such as, the desired throttle angle is calculated by the maximum of engine torque rather than an engine map and a simplified QP solving algorithm is used to reduce the computational time. Finally, this approach was implemented in a 16-bit microcontroller. The proposed switching logic uses zero as the threshold value, so it does not need calibration. Combined with the inverse longitudinal vehicle model and adaptive regulation of MPC, the switching logic can make use of the engine brake torque for various driving conditions and avoid high frequency oscillations automatically. The proposed approach can be easily applied to other vehicles because only three vehicle states and four vehicle parameters are needed. With this high precision speed tracking control system as a powerful tool, the planning and decision-making module of autonomous ground vehicles can focus on determining the desired path and the speed profile and make more sophisticated decisions, regardless of the neglected dynamics in the linear vehicle model.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant No. 51275041 and No. 91420203). The authors would like to thank the editors and the anonymous reviewers for their constructive comments and suggestions, which have greatly improved the presentation of this paper.

References

[1] H. Kim, D. Kim, I. Shu, K. Yi, Time-varying parameter adaptive vehicle speed control, IEEE Trans. Veh. Technol. 65 (2) (2016) 581–588.
[2] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, P. Mahoney, Stanley: the robot that won the DARPA grand challenge, J. Field Robot. 23 (9) (2006) 661–692.
[3] X. Li, Z. Sun, D. Cao, D. Liu, H. He, Development of a new integrated local trajectory planning and tracking control framework for autonomous ground vehicles, Mech. Syst. Signal Process. http://dx.doi.org/10.1016/j.ymssp.2015.10.021.
[4] X. Li, Z. Sun, Z. He, Q. Zhu, D. Liu, A practical trajectory planning framework for autonomous ground vehicles driving in urban environments, in: Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, Korea, 2015, pp. 1160–1166.
[5] H. Zhang, J. Wang, Vehicle lateral dynamics control through AFS/DYC and robust gain-scheduling approach, IEEE Trans. Veh. Technol. 65 (1) (2016) 489–494.
[6] A. Vahidi, A. Eskandarian, Research advances in intelligent collision avoidance and adaptive cruise control, IEEE Trans. Intell. Transp. Syst. 4 (3) (2003) 143–153.
[7] S. Moon, I. Moon, K. Yi, Design, tuning, and evaluation of a full-range adaptive cruise control system with collision avoidance, Control. Eng. Pract. 17 (4) (2009) 442–455.
[8] Y. Luo, T. Chen, K. Li, Multi-objective decoupling algorithm for active distance control of intelligent hybrid electric vehicle, Mech. Syst. Signal Process. 64–65 (2015) 29–45.
[9] D. Yanakiev, I. Kanellakopoulos, Speed tracking and vehicle follower control design for heavy-duty vehicles, Veh. Syst. Dyn. 25 (1996) 251–276.
[10] K.J. Hunt, J. Kallkkuhl, H. Fritz, T.A. Jonhansen, Th. Göttsche, Experimental comparison of nonlinear control strategies for vehicle speed control, in: Proceedings of International Conference on Control Application, Trieste, Italy, 1998, pp. 1006–1010.
[11] K.J. Hunt, T.A. Johansen, J. Kalluhl, H. Fritz, Th Göttsche, Speed control design for an experimental vehicle using a generalized gain scheduling approach, IEEE Trans. Control. Syst. Technol. 8 (3) (2000) 381–395.
[12] D. Anderson, Splined Speed Control Using SpAM (Speed-based Acceleration Maps) for an Autonomous Ground Vehicle Master's thesis, Virginia Polytechnic Institute and State University, USA, 2008.
[13] J. Wang, Z. Sun, X. Xu, D. Liu, J. Song, Y. Fang, Adaptive speed tracking control for autonomous land vehicles in all-terrain navigation: an experimental study, J. Field Robot. 30 (1) (2013) 102–128.
[14] F. Cabello, A. Acuna, P. Vallejos, M.E. Orchard, J.R. del Solar, Design and validation of a fuzzy longitudinal controller based on a vehicle dynamic simulator, in: Proceedings of the 9th IEEE International Conference on Control and Automation (ICCA), Santiago, Chile, 2011, pp. 997–1002.
[15] C.C. Tsai, S.M. Hsieh, C.T. Chen, Fuzzy longitudinal controller design and experimentation for adaptive cruise control and Stop&Go, J. Intell. Robot. Syst. 59 (2010) 167–189.

[16] J.E. Naranjo, C. Gonzalez, R. Garcia, T. de Pedro, Cooperative throttle and brake fuzzy control for ACC+Stop&Go maneuvers, IEEE Trans. Veh. Technol. 56 (4) (2007) 1623–1630.
[17] H. Zhang, J. Wang, State estimation of discrete-time Takagi–Sugeno fuzzy systems in a network environment, IEEE Trans. Cybern. 45 (8) (2015) 1525–1536.
[18] A. Murayama, M. Yamakita, Speed control of vehicles with variable valve lift engine by nonlinear MPC, in: Proceedings of ICROS-SICE International Joint Conference, Fukuoka, Japan, 2009, pp. 4128–4133.
[19] Y. Sakai, M. Kanai, M. Yamakita, Torque demand control by nonlinear MPC with constraints for vehicles with variable valve lift engine, in: Proceedings of IEEE International Conference on Control Applications, Yokohama, Japan, 2010, pp. 1642–1647.
[20] R. Rajamani, C. Zhu, Semi-autonomous adaptive cruise control systems, IEEE Trans. Veh. Technol. 51 (5) (2002) 1186–1192.
[21] J. Gerdes, K. Hedrick, Vehicle speed and spacing control via coordinated throttle and brake actuation, Control Eng. Pract. 5 (1997) 1607–1614.
[22] K. Yi, Y. Kwon, Vehicle-to-vehicle distance and speed control using an electronic-vacuum booster, JASE Rev. 4 (2001) 403–412.
[23] J. Zhou, H. Peng, Range policy of adaptive cruise control vehicle for improved flow stability and string stability, IEEE Trans. Intell. Transp. Syst. 6 (2) (2005) 229–237.
[24] S. Li, K. Li, R. Rajamani, J. Wang, Model predictive multi-objective vehicular adaptive cruise control, IEEE Trans. Control. Syst. Technol. 19 (3) (2011) 556–566.
[25] S. Eben, K. Li, J. Wang, Economy-oriented vehicle adaptive cruise control with coordinating multiple objectives function, Veh. Syst. Dyn. 51 (1) (2013) 1–17.

Min Zhu received B.S. and M.S. degrees in mechanical engineering from Chang'an University, Xi'an, China, in 2009 and 2012, respectively. He is currently pursuing a Ph.D. degree in mechanical engineering at Beijing Institute of Technology, Beijing, China. His research interests include autonomous ground vehicle control and automatic transmission.

Huiyan Chen received a Ph.D. degree in mechanical engineering from Beijing Institute of Technology, Beijing, China, in 2004. He is currently a Professor with the School of Mechanical Engineering and a Director of Intelligent Vehicle Research Center at Beijing Institute of Technology, Beijing, China. His research interests are autonomous ground vehicle control and automotive transmissions.

Guangming Xiong received a Ph.D. degree in mechanical engineering from Beijing Institute of Technology, Beijing, China, in 2005. He was a Visiting Scholar with the Department of Electrical and Computer Engineering, Brigham Young University, Provo, USA. He is currently an Associate Professor with the School of Mechanical Engineering, Beijing Institute of Technology, Beijing, China. His research interests include planning and control for autonomous ground vehicles and multi-robot system, human-machine interface and multi-vehicle cooperation.