 CMES, vol.116, no.1, pp.69-88, 2018

Distance Control Algorithm for Automobile Automatic Obstacle Avoidance and Cruise System

Jinguo Zhao[1, *]

Abstract: With the improvement of automobile ownership in recent years, the incidence of traffic accidents constantly increases and requirements on the security of automobiles become increasingly higher. As science and technology develops constantly, the development of automobile automatic obstacle avoidance and cruise system accelerates gradually, and the requirement on distance control becomes stricter. Automobile automatic obstacle avoidance and cruise system can determine the conditions of automobiles and roads using sensing technology, automatically adopt measures to control automobile after discovering road safety hazards, thus to reduce the incidence of traffic accidents. To prevent accidental collision of automobile which are installed with automatic obstacle avoidance and cruise system, active brake should be controlled during driving. This study put forward a neural network based proportional-integral-derivative (PID) control algorithm. The active brake of automobiles was effectively controlled using the system to keep the distance between automobiles. Moreover the algorithm was tested using professional automobile simulation platform. The results demonstrated that neural network based PID control algorithm can precisely and efficiently control the distance between two cars. This work provides a reference for the development of automobile automatic obstacle avoidance and cruise system.

Keywords: Obstacle avoidance and cruise, distance control, automobile, algorithm.

1 Introduction

With the development of society and the improvement of life quality, requirements on trips have become increasingly higher. Automobiles have gradually been a common means of transportation. The comfortability, security and environmental protection property of automobiles can affect the development of the automobile industry. The frequent traffic accidents in recent years make people pay more attentions to the security of automobile; as a result automobile automatic obstacle avoidance and cruise system emerges. Proportional-integral-derivative (PID) controller is a classic control method [Receanu (2013)], and its controlling performance was better than PI controller [Giwa (2016)]. Viknesh and Manikandan [Viknesh and Manikandan (2017)] studied the working principle of CSC converter using PID controller and obtained good results. Based on the analysis of adaptive cruise control (ACC) system, Gong et al. [Gong, Luo, Wang et al. (2010] put forward a parameter self-tuning fuzzy proportion integration

[1] Xijing University, No.1, Xijing Road, Chang'an district, Xi'an, Shaanxi, 710123, China.

[*] Corresponding Author: Jinguo Zhao. Email: jinguoz_vip@126.com

differentiation (PID) algorithm to design the control strategy of ACC; the parameters of PID controller could be adjusted online according to traffic conditions. The simulation results suggested that parameter self-tuning fuzzy PID controller which integrated the advantages of PID controller and fuzzy controller enhanced the driving safety and system response speed and provided drivers with a more comfortable driving experience.

Liang et al. [Liang, Zhao, Xiong et al. (2017)] proposed a parallel neural network PID (PNNPID) based acceleration control algorithm which overcame the defects of the traditional acceleration control algorithm in ACC such as slow system changes, poor dynamic performance and slow adjustment speed and developed a parallel control theory based automobile acceleration controller through analyzing the self-learning function of traditional serial neural network which could directly feedback errors. The experimental results suggested that serial neural network proportion integration differentiation (SNNPID) could operate within a maximum deviation of ±0.25 m/s, which was more accurate than PNNPID and featured by small average error, short regulation time and favorable transient performance. Kim et al. [Kim, Tomizuka and Cheng (2012)] proposed virtual automobile guidance model for automatic cruise system and combined it with linear secondary controller with variable weight to control and guide automobiles to keep a proper distance with the preceding automobile. Ploeg et al. [Ploeg, Scheepers, Nunen et al. (2011)] designed an automatic cruise system which integrated adaptive sensor and automobile communication and verified the feasibility of the system in short-distance automobile following. Lin et al. [Lin, Nguyen and Wang (2017)] applied a self-adaptive neural fuzzy predictive control in ACC system and tested its security and comfortability through simulation experiment. In this study, back-propagation (BP) neural network improved PID control algorithm was proposed to improve the preciseness of control on the speed and distance of automobile, and moreover the safe brake distance was deduced using a series of formulas. This work aims to improve the applicability of automobile automatic obstacle avoidance and cruise system in car following and distance control on expressway.

2 Automobile adaptive cruise control

2.1 System demand analysis

Automobile ACC system [Li (2013)] which can assist safe driving of automobiles has functions of automatic cruise and anticollision and can collect the surrounding conditions and the driving information of automobiles via radar sensor and automatically control driving according to different traffic conditions, which can improve the security and comfort of driving. Automobile ACC system is the direction of automobile manufacturing and will be extensively applied in automobiles in the future.

3 Control algorithm

3.1 PID algorithm

PID algorithm control [Liu, Bai and Ni (2011)] is an important component of ACC system, and calculus and proportion are the cores of PID algorithm. Fig. 1 shows the control principle of PID.

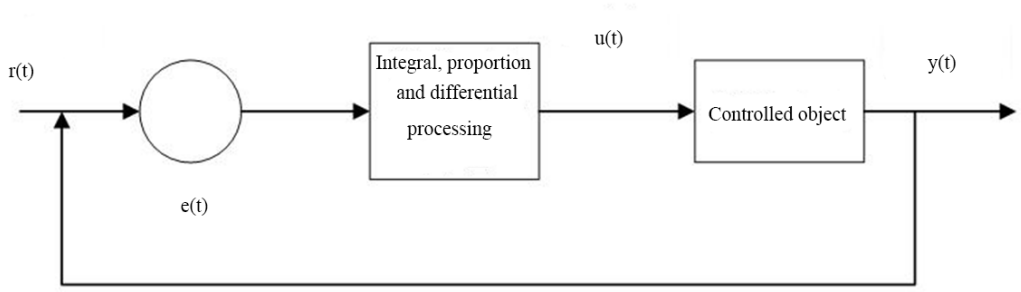


Figure 1: The principle of PID control

The formula was:

$$u(t) = K_p\left[e(t) + \frac{1}{K_I}\int_0^1 e(t)dt + K_D\frac{de(t)}{dt}\right], \quad (1)$$

where K_p stands for proportionality coefficient, K_I stands for integral time constant, and K_D stands for derivative time constant.

Generally signals transmitted in the internal system will meet a certain proportion. New signal e(t) will generate when there is a deviation in the proportion, and PID controller will correct it. With the increase of proportionality coefficient, the operation of system will accelerate, and the error generated can be reduced after rapid reflection of error. Integral control is used for improve preciseness of system control. The larger the integral time constant K_I is, the poorer the deviation cleaning ability is; the smaller the K_I is, the stronger the deviation cleaning ability is. Effective correction signal will be introduced in the early stage of the emergence of deviation signal if integral control is used. The larger the integral time constant K_d is, the smaller than adjustment time is. It is beneficial to the rapid correction of system in the early stage and improvement of system stability.

k refers to the sequence of sampling, and the increment of control object is:

$$\Delta u(k) = K_p\Delta e(k) + K_I e(k) + K_D[\Delta e(k) - \Delta e(k-1)], \quad (2)$$

where $\Delta e(k) = e(k) - e(k-1)$.

Fig. 2 shows the output features of different functions of controller.

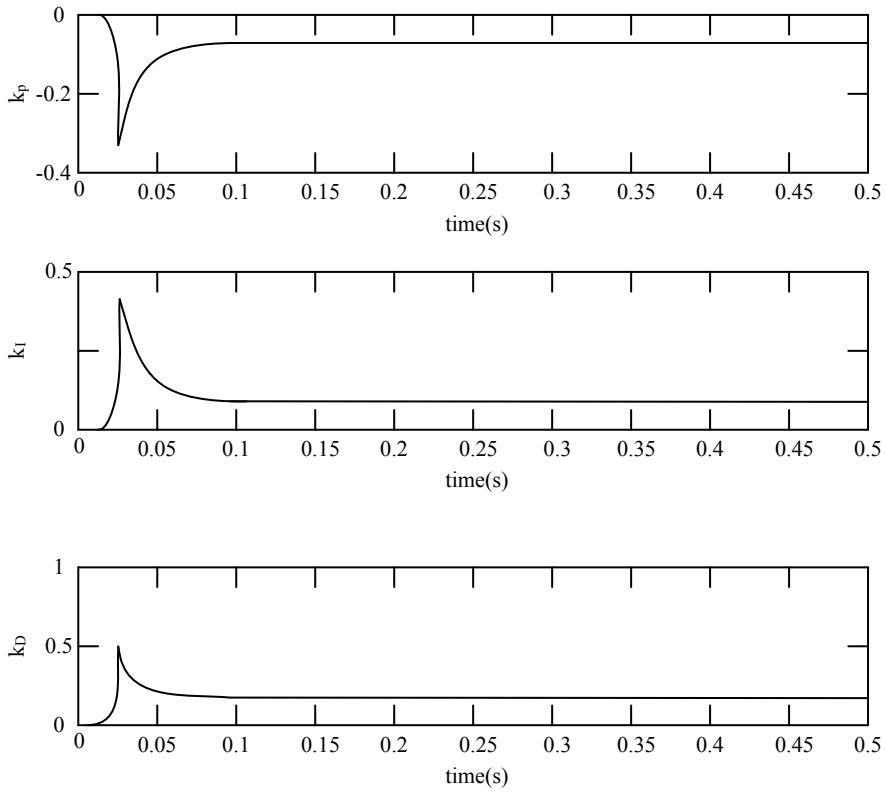


Figure 2: The output features of different functions of controller

It could be noted from Fig. 2 that controller can adjust in a short period of time and then keep stable when different parameters of controller have changes.

To further improve the performance, the traditional PID controller needs improvement.

3.2 Improved PID control based on BP neural network

3.2.1 BP based neural network algorithm

The input of network input layer is $Q_j = x(j), (j = 1,2,..., R)$. R meant that R signals need control. The value of R is set according to demands. x stands for neuron input. The input of the hidden layer of neural network is $net_i(k) = \sum_{j=1}^{R} w_{ij} Q_j$, where w_{ij} stands for the weighed proportional value of the hidden layer, and k stands for sampling sequence. The output of hidden layer is $Q_j(k) = f[net_i(k)], (j = 1,2,..., M)$. Sigmoid function is used in hidden layer as excitation function to represent the relationship between input nodes and output nodes.

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{3}$$

The input of the output layer of neural network is $net_i(k) = \sum_{i=1}^{M} w_{ij} Q_i(k)$; the output is

$$Q_l(k) = g(net_l(k))(l = 1,2,3), Q_1(k) = K_p, Q_2(k) = K_I, Q_3(k) = K_D \tag{4}$$

The relational expression for the transmission and association of signals in transition layer and hidden layer is:

$$x(k) = f\{K_r, x_c(k) + K_I[u(k-1)]\}, \tag{5}$$

K_r stands for the weighed proportional value of transition layer.

3.2.2 Learning algorithm

A relatively mature BP neural network needs a matched learning algorithm which can correct the weighed proportional value of neuron. Ideal learning is to compare the output results of neural network with supervised learning output results; a large deviation means that weighed [Chen and Pi (2014)] proportional value cannot satisfy control requirements and neural network needs to be adjusted according to deviation. Non-ideal learning means comparing the output results in different routes which are input with the same signal, adjust weighed proportion value to make the output values in different routes same. Fig. 3 shows the application of learning algorithm in control system.

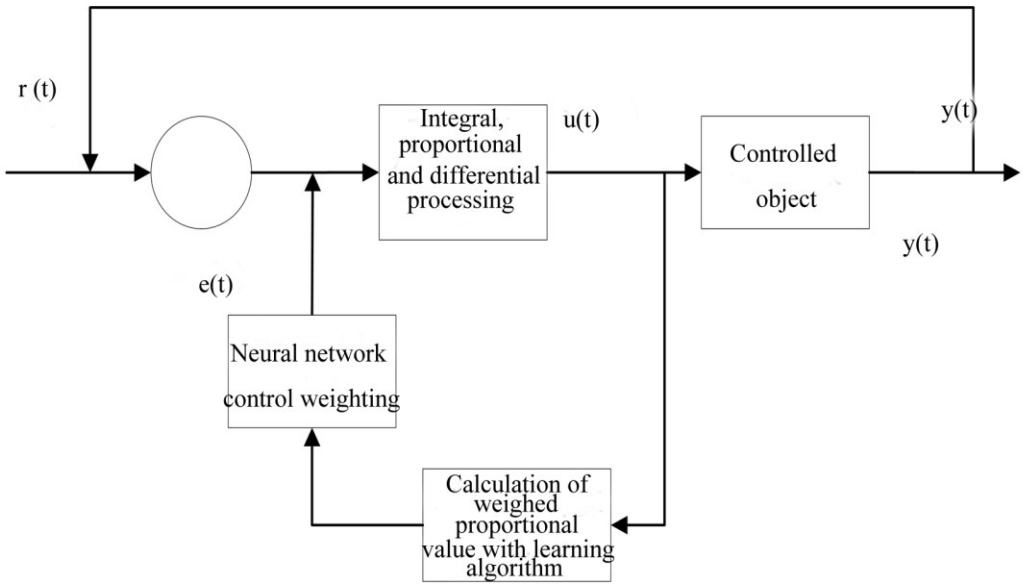


Figure 3: The learning procedure of control algorithm

(1) Supervised learning algorithm

Firstly the minimum weighed proportional value of neuron [Budak, Senger, Guo et al. (2017)] $w_j(0)(j = 0,1,\ldots, n)$ was taken and it should not be equal to 0. Then it was supposed that there were p groups of samples, the input signal was $u_p = (u_{0p}, u_{1p}, \ldots u_{np}), (p = 1,2,\ldots, L)$, and the output signal was $d_p (p = 1,2,\ldots L)$. After assignment,

$$d_p = \begin{cases} +1, u_p \in A \\ -1, u_p \in B \end{cases}. \tag{6}$$

The output signal of the perceptron of supervised learning algorithm was:

$$y_p(t) = f\left[\sum_{j=0}^{m} w_j(t) u_{jp}\right]. \tag{7}$$

The weighed proportional value was:

$$w_i(t+1) = w_i(t) + \eta\left[-d_p - y_p(t)\right]u_{jp}, 0 < \eta \le 1. \tag{8}$$

The output of BP neural network was:

$$y_{ip} = f\left[\sum_{j} w_{ij}(t) I_{jp}\right], \tag{9}$$

where I_{jp} stands for the i-th neuron of the j-th input of the p-th group.

(2) Ideal learning algorithm

Learning algorithm was obtained by combining the non-ideal results with the ideal results.

$$\Delta w_{ij}(k) = w_{ij}(k+1) - w_{ij}(k) = \sigma\left[d_j(k) - q_j(k)\right]q_j(k)q_i(k), \tag{10}$$

where q_i stands for the effect value of neuron i, q_j stands for the effect value of neuron j, and w_{ij} stands for the interaction between neuron i and j.

3.2.3 The procedures of PID control optimization with BP neural network

Firstly the number of layers and neurons of BP neural network [Ma, Yao and Wang 2017] was confirmed. Then the weighed proportion of neurons, $w_{ij}(0)$ and $w_{li}(0)$, were confirmed. Then learning rate σ [Kaminski and Orlowska-Kowalska (2015)] and smooth factor α [Wu and Moody (2014)] were set; k was set as 1. Some initial values were substituted to calculate the initial deviation of the output value and ideal value, $e(k) = r(k) - y(k)$. The deviation of the ideal output value was $e(i)(i = k, k-1, \ldots k-p)$, and it was taken as the input signal of learning algorithm.

Next the output values of neural network $K_p(k)$, $K_I(k)$ and $K_D(k)$ were regarded as the parameters of PID controller. u(k) which was obtained after calculation was taken as the control signal of learning algorithm.

Weighed proportional values $w_{li}(k)$ and $w_{ij}(k)$ were calculated out using learning algorithm. The weighed proportional values were adjusted constantly till satisfying the control requirements.

3.2.4 Comparison between the performance of the improved algorithm and PID algorithm

It could be noted from Tabs. 1 and 2 that the output of PID algorithm was unstable and keep going up and down from 0 s to 30 s though the fluctuation became smaller gradually, indicating the low control preciseness of PID algorithm; but the improved PID algorithm controlled the output in less than one second, and the output showed no changes afterwards. Compared to PID algorithm, the improved PID algorithm performed better in eliminating interference and stabilized output in a shorter time, indicating that it could more effectively enhance control preciseness.

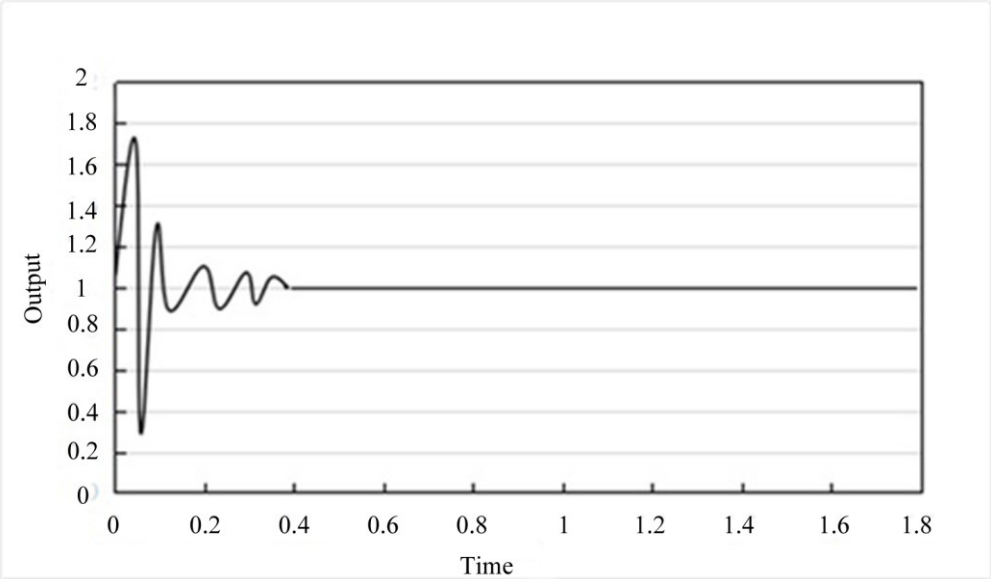
Table 1: Control of the improved PID algorithm on variable output

Time (s)	0	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8
Output	1	1.03	1	1	1	1	1	1	1	1

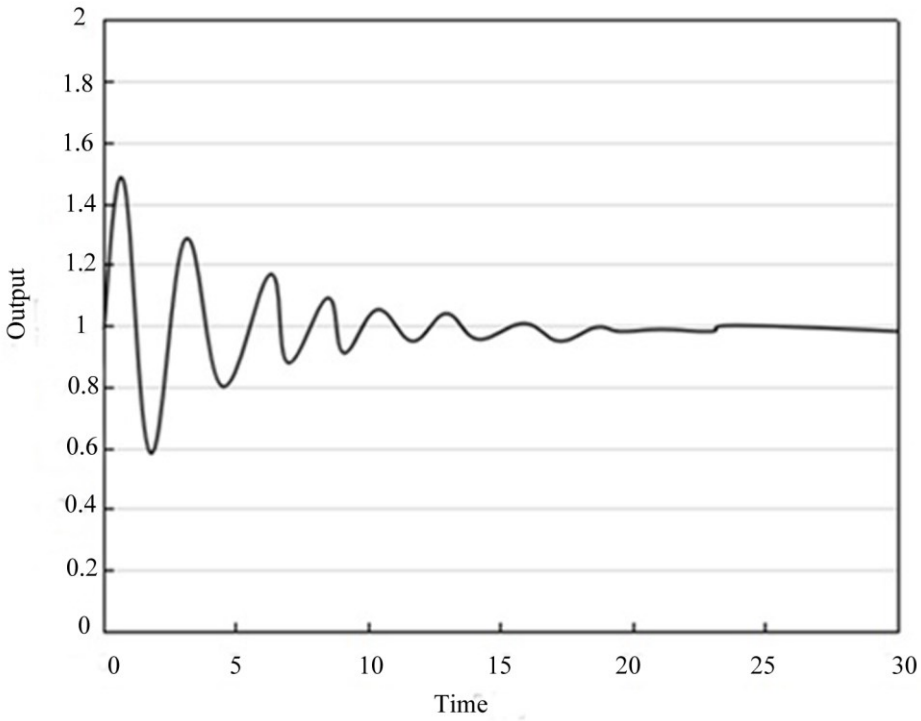
Table 2: Control of PID algorithm on variable control

Time (s)	0	5	10	15	20	25	30
Output	1	0.8	0.97	0.94	0.99	1	1

After the improved algorithm enhanced the control preciseness of the system, automobile distance and speed needed to be deduced and calculated.



A. The control variable output of the improved algorithm



B. The control variable output of PID algorithm

Figure 3: The curves of control variable output of the two algorithms

3.3 Adaptive distance control

3.3.1 Control of automobile speed

Dynamically stable equation could be obtained according to the force equilibrium relationship during steady driving [Cranmer, Shahbakhti and Hedrick (2012)]. Automobile driving force was:

$$F_t = F_f + F_w + F_i = \frac{T_t}{r} = \frac{T_{eq} i_g i_0 \eta_t}{r} = G\cos\theta\left[f_0 + f_1\left(\frac{v}{100}\right) + f_4\left(\frac{v}{100}\right)\right] + \frac{C_D A v^2}{21.15} + G\sin\theta, \tag{11}$$

where F_t stands for automobile driving force, F_f stands for rolling resistance, F_w stands for air resistance, F_i stands for slope resistance, T_{eq} stands for engine torque (unit: N·m), i_0 stands for final driver ratio, η_t stands for the mechanical efficiency of driver, r stands for the radius of wheels (unit: m), f_0, f_1 and f_4 stand for rolling resistance coefficients, v stands for the driving speed of automobile, G stands for the gravity of automobile, C_D stands for air resistance coefficient, and A stands for the windward area of automobile body.

The followings were obtained according to Eq. (11).

$$T_{tq} = \frac{r_s}{i_g i_0 \eta_T} \tag{12}$$

$$\left\{G\cos\theta\left[f_0 + f_1\left(\frac{v}{100}\right) + f_4\left(\frac{v}{100}\right)\right] + \frac{C_D A v^2}{21.15} + G\sin\theta\right\} \tag{13}$$

Air resistance was not considered in this study, and moreover rolling resistance was replaced by sliding resistance. Then engine torque could be expressed as:

$$T_{tq} = \frac{rG\left(f_0\cos\theta + \sin\theta\right)}{i_g i_0 \eta_r} \tag{14}$$

The automobile cruise control system controlled automobile speed by comparing the actual engine torque T_C with the theoretical engine torque. When $T_C > T_{eq}$, the actual engine torque of automobile was excessively large; the speed of the cruising automobile exceeded the preceding automobile and they gradually approached. When the preceding automobile slowed down and braked, the cruising automobile might face with collision and emergency brake, which could not satisfy the requirements of automatic cruising. $T_C = T_{eq}$ indicated that the cruising automobile could follow the preceding automobile in a preset speed. When $T_C < T_{eq}$, the actual engine torque was smaller than the theoretical torque, which could not satisfy the requirements of constant speed cruise.

3.3.2 Automobile braking distance

Analysis on the braking distance of automobile is the basis for control on the distance between the cruising and preceding automobiles. Automobile braking can be divided into

four stages. t_1 refers to the nerve reaction time of the driver [Hwakyung and Hocheol (2012)], when the system alarms, usually 0.3~1.2. t_2 refers to the time that the driver moved his foot from accelerator pedal to brake pedal to overcome the resistance produced by braking pedal. In the period, the automobile will produce a resistance and acceleration, but they can be ignored as they are too small; the automobile runs in the original speed. t_3 refers to the time from linear increase of braking force to the maximum braking acceleration. t_4 refers to the time from braking at the maximum acceleration to stop of automobile. The whole process is shown in Fig. 4.

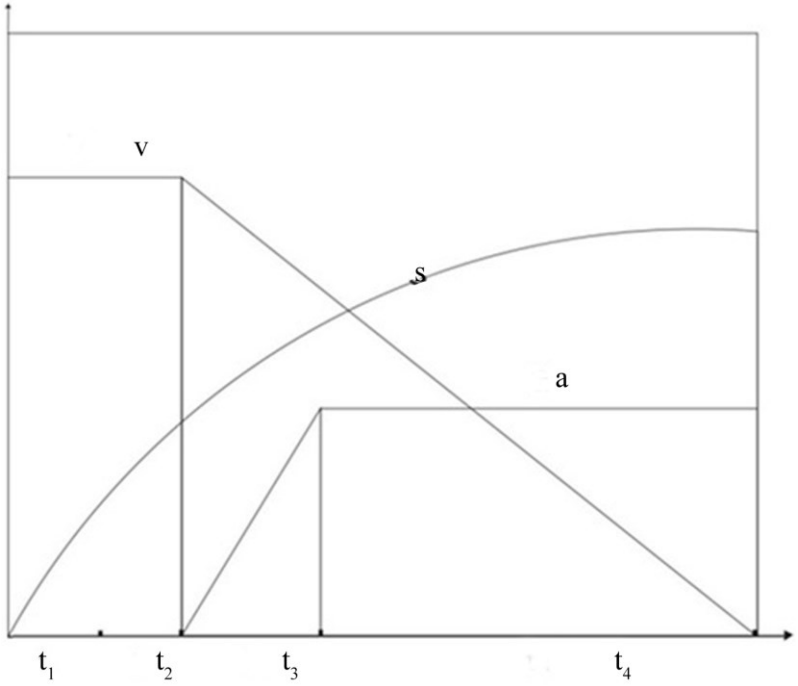


Figure 4: The curves for the time-varying speed v, driving distance s and braking acceleration a of the cruising automobile

(1) Driving distance of automobile in t_1 and t_2

The cruising speed of the cruising automobile was set as v_1, and then the computational formula for the driving distance in t_1 and t_2 was:

$$s_1 = v_1(t_1 + t_2) . \tag{15}$$

(2) Driving distance of automobile in t_3

In t_3, the braking force of the cruising automobile increased linearly, from 0 to the maximum value. The maximum acceleration of automobile braking a_{max} was obtained. The real-time automobile speed when reaching the maximum braking acceleration was set as v_2, and the driving distance was set as s_2. Then the computational formulas were:

$$v_2 = v_1 - \frac{a_{\max} t_3}{2}, \tag{16}$$

$$s_2 = v_1 t_3 - \frac{a_{\max}}{6} t_3^2. \tag{17}$$

(3) Driving distance of automobile in t_4

In t_1, automobile slowed down at the maximum braking acceleration $a_{\max}$ till stopped. The driving distance was set as s_3, then

$$s_3 = \frac{v_1^2}{2a_{\max}} - v_1 \frac{t_3}{2} + \frac{a_{\max}}{8} t_3^2. \tag{18}$$

Therefore the braking distance was:

$$s_0 = v_1\left(t_1 + t_2 + \frac{t_3}{2}\right) + \frac{v_1^2}{2a_{\max}} - \frac{a_{\max} t_3^2}{24}. \tag{19}$$

t_3^2 could be ignored as the value was too small, then the formula was simplified as:

$$s_0 = v_1\left(t_1 + t_2 + \frac{t_3}{2}\right) + \frac{v_1^2}{2a_{\max}}. \tag{20}$$

3.3.3 Braking warning distance

Braking warning distance referred to the driving distance from the brake device took effect to automobile stopped. Generally there were extreme situation and normal situation.

In extreme situation, i.e. the preceding automobile suddenly stopped, the braking warning distance of the cruising automobile was:

$$d_b = v_1\left(t_2 + \frac{t_3}{2}\right) + \frac{v_1^2}{2a_{\max}} + d_0, \tag{21}$$

where d_0 stands for the minimum safe distance between the cruising and preceding automobiles after braking, usually 2~5 m.

In normal situation, i.e., the preceding automobile slowed down gradually till stopped. The speed of the preceding automobile was $v_{\text{preceding}}$, and the computational formula for braking warning distance when the braking acceleration was a_2 was:

$$d_b' = \frac{1}{2}\left(\frac{v_1^2}{a_{\max}} - \frac{v_{\text{preceding}}^2}{a_2}\right) + v_1 t_1 + d_0 \quad (22)$$

3.3.4 Reminding and warning distance

Reminding and warning distance was the sum of braking warning distance and driving distance during the reaction time of driver. There were extreme situation and normal situation.

In extreme situation, reminding and warning distance was

$$d_t = v_1\left(t_1 + t_2 + \frac{t_3}{2}\right) + \frac{v_1^2}{2a_{\max}} + d_0 \quad (23)$$

In normal situation, the computational formula of reminding and warning distance was:

$$d_t' = \frac{1}{2}\left(\frac{v_1^2}{a_{\max}} - \frac{v_{\text{preceding}}^2}{a_2}\right) + v_1(t_1 + t_2) + d_0 \quad (24)$$

3.4 Procedures of automatic constant-speed cruising and distance control

In this study, automobile control was realized using BP neural network improved PID control algorithm. Firstly the deviation between input speed signal and actual driving speed was calculated; then the operation result was input into the cruising automobile to control speed, and moreover real-time control was performed using sensor feedback. Finally the difference with the actual speed was controlled to approach to 0. The system control flow is shown in Fig. 5.

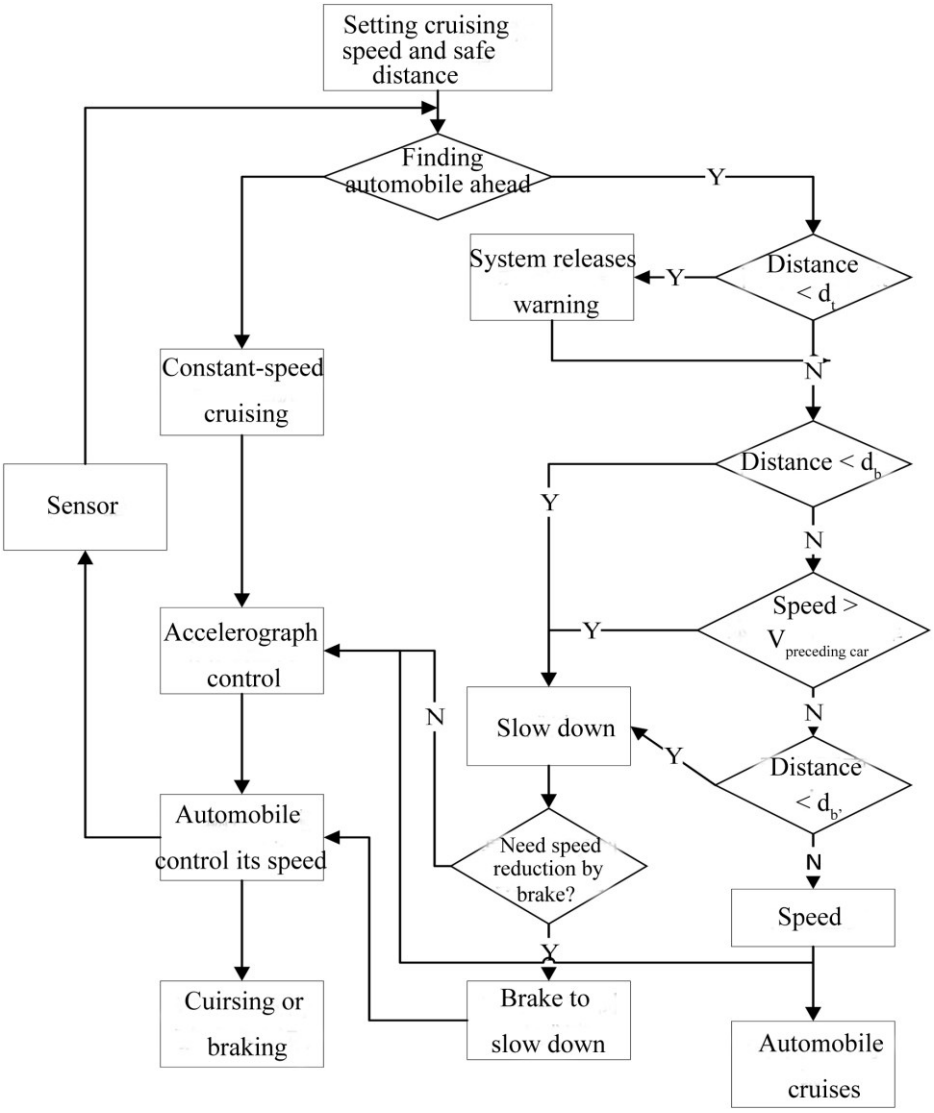


Figure 5: The flow of constant-speed cursing and distance control of the automobile cruising system

4 Simulation experiment

The automobile obstacle avoidance and cruise system was tested using simulation experiment. Simulink and Car Sim software were used [Laarabi and Bruno (2016)]. Simulink software with strong functions can support system-level design and realize dynamic modeling and simulation analysis. Car Sim software which is for vehicle dynamics can analyze the performance of automobiles such as stability and braking. The system was mainly applicable to expressway.

Tab. 3 shows the specific setting of some parameters.

Table 3: The determination on the values of relevant parameters in simulation experiment

Model parameters	Symbol	Numerical	Unit
Total weight	m	1370	kg
Transmission efficiency of transmission system	η	0.9	-
Rolling radius of wheels	r	0.335	m
Tread	d	1.55	m
Air resistance coefficient	C_D	0.342	-
Maximum acceleration of braking	$a_{\max}$	6	m/s
Response time of driver	t_1	1.2	s
Coordination time of brake	t_2	0.2	s
Minimum safe distance between two automobiles	d_0	5	m
Proportional coefficient	K_p	100~450	-
Integral time constant	K_I	10~45	-
Derivative time constant	K_I	1~10	-

4.1 Simulation results of automobile speed control

The initial speed of automobile was set as 60 km/h, and the expectant cruising speed was 70 km/h. The system control result is shown in Fig. 6.

It could be noted from Fig. 6 that the response of the system was fast and moreover it had a high stability. To further study the performance of the system, the traditional PI control method was compared with the method developed in this study. The experimental results are shown in Tabs. 4, 5 and 6.

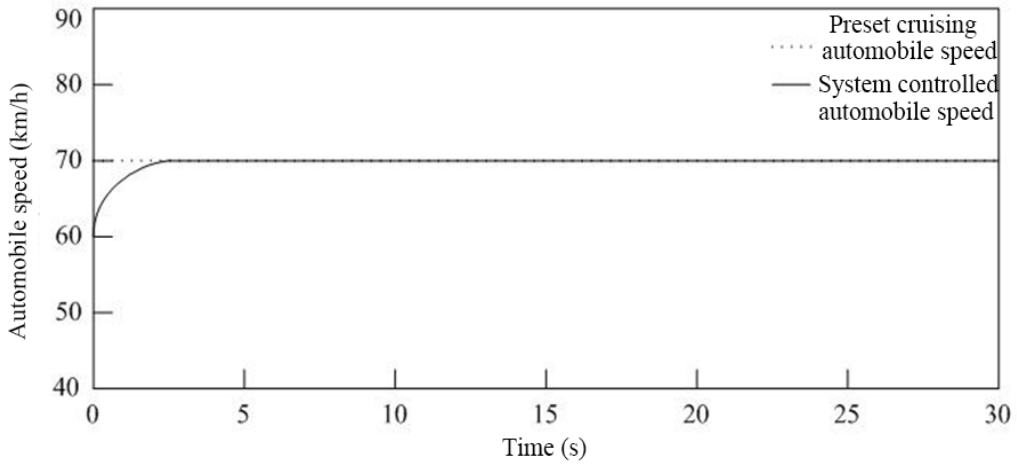


Figure 6: The control result of automobile speed

Table 4: The experimental results of PI system

Setting speed km/h	30.03	40.46	50.27	60.31	70.40	80.46	90.13	100.11	110.02	120
Actual speed km/h	31.26	41.08	51.36	61.32	70.98	82.12	91.64	98.79	108.28	118.97

Table 5: The experimental results of the system developed in this study

Setting speed (km/h)	30.03	40.46	50.27	60.31	70.40	80.46	90.13	100.11	110.02	120
Actual speed (km/h)	30.34	40.78	50.47	60.11	70.19	80.15	90.13	99.89	109.84	119.75

Table 6: The comparison of the error rate between the two systems

Control error of the system developed in this study%	1.02	0.78	0.40	-0.33	-0.30	-0.39	0	-0.22	-0.16	-0.21
Control error of PI system%	4.10	1.53	2.17	1.67	0.82	2.06	1.68	-1.32	-1.58	-0.86

Tab. 3 suggests that the control error at the constant speed mode was between -0.40% and 1.02%. Hence it was concluded that the controller could accurately control automobile speed and control the error within a small scope, which could satisfy the requirement of precise control for automobile speed. The comparison of the experimental results showed that the controller proposed in this study had smaller error, suggesting a better performance.

4.2 Simulation results of automobile distance control

The driving speed of automobile was set as 70 km/h, and the current driving speed of automobile was 60 km/h. It had a distance of 45 m with the automobile in front. When the speed slowed down, the control result of automobile distance is shown in Fig. 7.

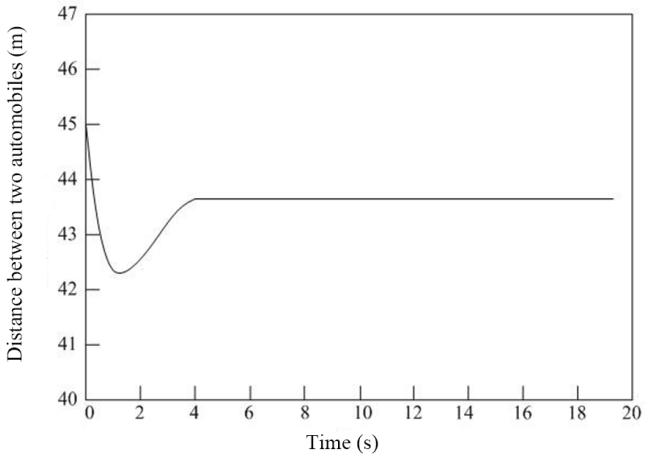


Figure 7: The control result of automobile distance

It could be noted from Fig. 7 that the system designed in this study could rapidly control the automobile distance and respond in a short period of time. To further study the performances of the systems in controlling automobile distance control, the following experiments were made.

Tab. 7 demonstrated that, when the speed of the cruising automobile was set as 120 km/h and the safe automobile distance was set as 108 m, the error between the speed was between -0.82% and 0.52% and the error of automobile distance was between -0.47% and 0.63%; when the speed of the cruising automobile was set as 110 km/h and the safe automobile distance was set as 108 m, the error between the speed was between -0.74% and 0.56% and the error of automobile distance was between -0.13% and 0.18%; when the speed of the cruising automobile was set as 100 km/h and the safe automobile distance was set as 108 m, the error between the speed was between -0.43% and 0.30% and the error of automobile distance was between -0.06% and 0.10%; when the speed of the cruising automobile was set as 80 km/h and the safe automobile distance was set as 108 m, the error between the speed was between -1.21% and 2.11% and the error of automobile distance was between -0.41% and 0.22%. In general, the error between the speed was between -1.21% and 2.11% and the error between the automobile distance was between -0.47% and 0.63%. The improved automatic cruising system had high control

preciseness and small error and was applicable to following and distance control on expressway.

Table 7: The results of the simulation experiment for automobile cruising and following distance control

The cruising speed was set as 120 km/s; it follows the preceding automobile; the safe following distance was set as 108 m				
Speed of the preceding automobile (km/h)	110.89	100.14	90.41	80.57
Speed of the cruising automobile (km/h)	110..77	100.25	89.67	80.99
Actual automobile distance (m)	107.96	107.49	108.68	107.69
Error of automobile speed%	-0.17	0.11	-0.82	0.52
Error of automobile distance %	0.04	-0.47	0.63	-0.29
The cruising speed was set as 110 km/s; it follows the preceding automobile; the safe following distance was set as 108 m				
Speed of the preceding automobile (km/h)	100.48	90.32	80.14	70.76
Speed of the cruising automobile (km/h)	99.74	90.07	80.59	70.61
Actual automobile distance (m)	108.19	108.14	107.86	108.12
Error of automobile speed%	-0.74	-0.28	0.56	-0.21

Error of automobile distance %	0.18	0.13	-0.13	0.11
The cruising speed was set as 100 km/s; it follows the preceding automobile; the safe following distance was set as 108 m				
Speed of the preceding automobile (km/h)	90.49	80.78	70.13	60.52
Speed of the cruising automobile (km/h)	90.11	80.54	69.83	60.34
Actual automobile distance (m)	108.10	108.09	108.11	108.06
Error of automobile speed%	-0.42	-0.30	-0.43	0.30
Error of automobile distance %	0.09	0.08	0.10	0.06
The cruising speed was set as 80 km/s; it follows the preceding automobile; the safe following distance was set as 108 m				
Speed of the preceding automobile (km/h)	70.53	60.16	50.79	40.82
Speed of the cruising automobile (km/h)	70.64	59.43	50.47	41.68
Actual automobile distance (m)	107.89	108.24	108.11	107.56
Error of automobile speed%	0.16	-1.21	-0.64	2.11
Error of automobile distance %	-0.10	0.22	0.10	-0.41

5 Conclusion

Study on automobile automatic cruising system is a hot spot in recent years, which is of great significance to road transportation safety. This study put forward BP neural network improved PID control algorithm to control automobile cruising system, which made up the defects of PID algorithm and moreover enhance the control preciseness of control system. The actual motor engine speed was calculated based on driving equilibrium to determine whether the actual speed of automobile could satisfy cruising requirement. The automobile braking process was also analyzed to deduce the calculation method for braking distance. Moreover braking warning distance and reminding and warning distance were proposed, which could help the control of braking and distance control. Finally simulation experiment was used to test the automobile automatic obstacle avoidance and cruise system. The results demonstrated that the automatic obstacle avoidance and cruise system could be applicable to cruising on expressway and keep a safe distance with the preceding automobile and had small errors in controlling following and constant-speed cruising. The system provides an approach for the further implementation of road transportation safety. The system needs to be further improved in practical application.

Acknowledgement: This study was supported by the Scientific Research Project of Education Department of Shaanxi (15JK2186).

References

Budak, U.; Senger, A.; Guo, Y.; Akbulut, Y. (2017): A novel microaneurysms detection approach based on convolutional neura l networks with reinforcement sample learning algorithm. *Health Information Science & Systems*, vol. 5, pp. 14.

Chen, X.; Pi, H. (2014): The study on neural network feedback based on learning algorithm. *Proceedings of the 2012 International Conference on Cybernetics and Informatics*.

Cranmer, A.; Shahbakhti, M.; Hedrick, J. K. (2012): Grey-box modeling architectures for rotational dynamic control in automotive engines. *American Control Conference*, vol. 50, no. 6, pp. 1278-1283.

Giwa, A. (2016): PI and PID control of a fuel additive reactive distillation process. *Journal of Engineering & Applied Sciences*, vol. 11, no. 11, pp. 6779-6793.

Gong, L.; Luo, L.; Wang, H.; Liu, H. (2010): Adaptive cruise control design based on fuzzy-PID. *International Conference on E-Product E-Service and E-Entertainment*, pp. 1-4.

Hwakyung, S.; Hocheol, L. (2012): Characteristics of driving reaction time of elderly drivers in the brake pedal task. *Journal of Physical Therapy Science*, vol. 24, no. 7, pp. 567-570.

Kaminski, M.; Orlowska-Kowalska, T. (2015): An on-line trained neural controller with a fuzzy learning rate of the Levenberg-Marquardt algorithm for speed control of an electrical drive with an elastic joint. *Applied Soft Computing*, vol. 32, pp. 509-517.

Kim, S. G.; Tomizuka, M., K.; Cheng, H. (2012): Smooth motion control of the adaptive cruise control system by a virtual lead vehicle. *International Journal of*

Automotive Technology, vol. 13, no.1, pp. 77-85.

Laarabi, M. H.; Bruno, R. (2016): A generic software framework for carsharing modelling based on a large-scale multi-agent traffic simulation platform. *International Workshop Agent Based Modelling Urban Systems*, pp. 88-111.

Li, S. E. (2013): Economy-oriented vehicle adaptive cruise control with coordinating multiple objectives function. *Vehicle System Dynamics*, vol. 51, pp. 1-17.

Liang, J.; Zhao, T.; Xiong, X.; Zhu, N. (2017): Design of vehicle acceleration controller based on parallel neural network PID. *Journal of Southwest Jiaotong University*, vol. 52, pp. 626-632.

Lin, Y. C.; Nguyen, H. L. T.; Wang, C. H. (2017): Adaptive neuro-fuzzy predictive control for design of adaptive cruise control system. *International Conference on Networking, Sensing and Control*, pp. 767-772.

Liu, K.; Bai, M.; Ni, Y. (2011): Two-wheel self-balanced car based on Kalman filtering and PID algorithm. *International Conference on Industrial Engineering and Engineering Management*, pp. 281-285.

Ma, L.; Yao Y.; Wang, M. (2016): The optimizing design of wheeled robot tracking system by PID control algorithm based on BP neural network. *International Conference on Industrial Informatics-Computing Technology, Intelligent Technology, Industrial Information Integration*, pp. 34-39.

Ploeg, J.; Scheepers, B. T. M.; Nunen, E. V.; Wouw, N. V. D.; Nijmeijer, H. (2011): Design and experimental evaluation of cooperative adaptive cruise control. *International IEEE Conference on Intelligent Transportation Systems*, vol. 32, no. 14, pp. 260-265.

Receanu, D. (2013): Modeling and simulation of the nonlinear computed torque control in simulink/MATLAB for an industrial robot. *SL*, vol. 10, no. 2, pp. 95-106.

Viknesh, T. V.; Manikandan V. (2017): Modeling of canonical switching cell converter using genetic algorithm. *Computer Modeling in Engineering & Sciences*, vol. 113, no. 1, pp. 109-116.

Wu, L.; Moody, J. (2014): A smoothing regularizer for recurrent neural networks. *Neural Computation*, vol. 8, no. 3, pp. 461-489.