Particle Swarm Optimization Algorithm for Regenerative Braking Fuzzy Control of Electric Vehicle

Liao Qin *

Institute of Automatic, Chongqing University of Posts and Telecommunications, 400065, China

albert8964@163.com

Keywords: Regenerative braking, fuzzy controller, particle swarm optimization, electric vehicle.

Abstract. Improving raking energy regeneration efficiency is a vital problem of electric vehicle. Particle swarm optimization is introduced for regenerative braking fore distribution fuzzy controller, using membership functions and rules of fuzzy controller as optimization object and using limit of input as constraint condition. In this article, based on the front and rear braking force distribution strategy, a traditional fuzzy controller is designed. Then we show how to use particle swarm optimization algorithm to optimize it. Compared to the traditional one, we carry on some simulations in ADVISOR software. The results show that, the braking torque is improved and the braking energy regeneration efficiency raises by 7.19 percent, which indicates the validity of the proposed fuzzy controller.

Introduction

Because of the advantages of electric vehicle (EV), such as high efficiency, low noise and zero release, it has strength on energy conservation and environmental protection, but there are two important problems which are improving storage ability and utilization efficiency of energy. The difference between EV and traditional vehicle is the EV could realize regenerative breaking, transforming some kinetic energy into other forms to be stored and used. While nowadays On-board energy technology hasn’t developed, improving energy regenerative efficiency definitely has huge meaningful.

The EV would coupling with friction brake when regenerative breaking. In order to solve this, the paper [1],[2] use fuzzy control’s advantages, such as good robustness, short response time and fitness of the nonlinear system to control and coordinate regenerative breaking and friction brake to get good energy recovery effect and brake stability; but electrical machine is limited by Battery charge rate and the characteristics of electrical machine.

As a global optimization method based on swarm intelligence, particle swarm optimization (PSO) has simple iterative process, and need not operations of crossover and mutation. Thus, PSO can improve the design of the fuzzy controller [3]. In this article, using PSO algorithm and treating braking energy regeneration efficiency as a goal, to optimize fuzzy controller of the regenerative braking force distribution. Finally, simulations are taken on the drive cycle of UDDS, comparing with traditional fuzzy controller, the results show the validity of the proposed method.

Fuzzy controller for regenerative braking force distribution

Considering the influence factor of regenerative braking,a fuzzy controller is designed which has three inputs and one output. The inputs of the fuzzy controller represent the velocity of the electric vehicle v, the braking strength and the State-of-Charge of the battery SOC , respectively; the output is the regenerative braking force scale factor K,which represents motor braking torque account for the proportion of the maximum of the available braking force.

The membership functions (MF) on the universes of discourses and linguistic values for three inputs and one output are shown in Fig.1. The linguistic values “L, M, H” represent “low, middle, high”, respectively. The rule-base can be treated as a queue of fuzzy sets of input and output variables.

A typical rule will take on the form: “If z is L, v is L and SOC is L, then K is L”. Table 1 shows the premises and consequents of all the initial rules.

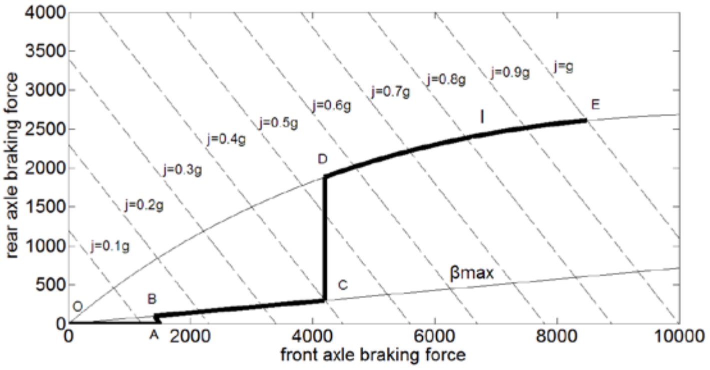

Fig.1 front and rear braking force distribution

Table 1 initial rules of fuzzy controller

K	Velocity v					
The braking strength z		L	M	H		SOC
	L	L	M	M	L	
	M	L	M	M	L	
	H	L	L	L	L	
	L	L	H	H	M	
	M	L	H	H	M	
	H	L	L	L	M	
	L	L	L	L	H	
	M	L	L	L	H	
	H	L	L	L	H	

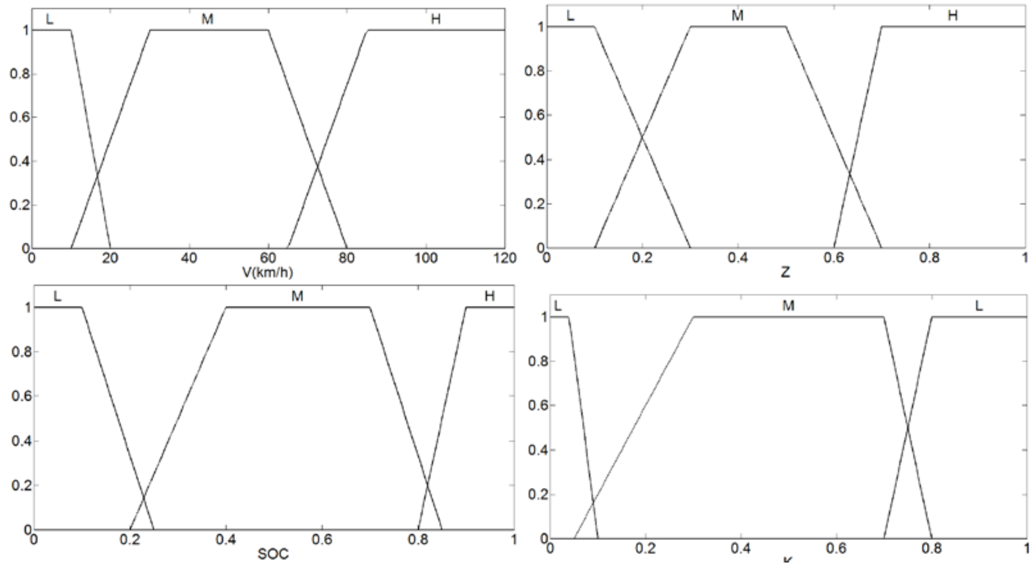

Fig.2 Membership functions of v, z, SOC and K

Fuzzy Controller Based PSO for regenerative braking force distribution

The MFs and rules of fuzzy controller relies too much on the expert experience, and is hard to get the optimal performance. Due to the non-linear of fuzzy control, it is difficult to optimize the controller with gradient method. While PSO is a global optimization method based on swarm intelligence, and can be effective in complex space search. Therefore, the article uses PSO to optimize the fuzzy controller to get optimal control effect.

PSO algorithm. In PSO, each particle of the swarm flies in a d-dimension space, and the position at a certain instant is identified by the vector of the coordinates x each x represents a parameter of the problem that has to be optimized. First, each particle is randomly located at a position, and moves with a random velocity. Then, aimed at the optimization problem, each particle determines a fitness value in every iteration. Finally, each particle searches for the optical solution by updating two extreme values: P_{id} is the best position found by the agent during its own wandering up to a $i-th$; P_{gd} is the best position found by the entire swarm[4]. Particle updates velocity and position according to:

$$\begin{cases} v_{id}(t+1)=\omega v_{id}(t)+c_1 r_1[P_{id}-x_{id}(t)]+c_2 r_2[P_{gd}-x_{id}(t)] \\ x_{id}(t+1)=x_{id}(t)+v_{id}(t+1) \end{cases}$$

v is the velocity of particle i in $i-th$ generation in d-dimension component, and x is the position; c_1 and c_2 are the cognitive and the social rate, respectively. The value of c_1 encourages the independent search for the best, regardless of the experience of the swarm; c_2 controls the exploitation of the actual best. ω is the inertia weight. When the value of ω is larger, the global search ability of PSO is stronger; when it is less, the local search ability of PSO is better. Thus, ω decreases linearly as the generation of iteration reducing, according to:

$$\omega=\omega_{\max}-k\frac{\omega_{\max}-\omega_{\min}}{k_{\max}}$$

ω is the total generation of iteration; $\omega_{\max}$ and $\omega_{\min}$ are the maximum weight and the minimum weight, respectively. In each iteration, algorithm updates the value of P_{id} and P_{gd} based on the fitness of particle.

Fuzzy controller based on PSO algorithm. As far as the MF of v is concerned, it consists of a number of trapezoids. In order to simplify the computation complexity, and reduce the dimension of the parameters code, the two top intersection points of the trapezoidal MF are encoded. While the top and bottom boundary of MF determined by some constraint conditions and design principles of fuzzy controller. The peak value has depicted in Fig.3. From the diagram, we can see that, x_1, x_2, x_3, x_4 are the parameters that need to be optimized in MF of v. The set of optimized parameters in MFs of SOC, z, K is also similar to that in v. Thus, the number if optimized parameters is 16.

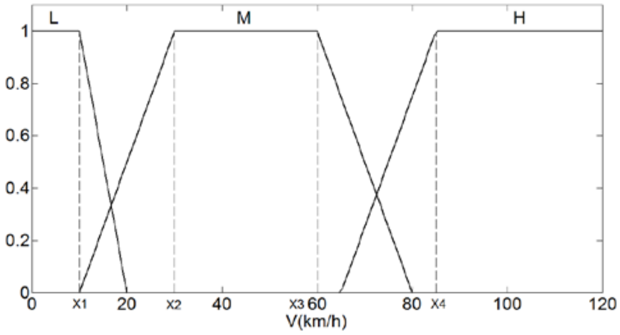

Fig.3 the optimized parameters of MF

Table 2 result of orthogonal test

1	2	…	7	8	9	10	…	26	27
x_1	x_2	…	x_7	1	1	1	…	3	3

In the case that the queue of premises of rules is given, we just need to optimize the corresponding consequents. From Table 1, we can see the number of regenerative braking force distribution fuzzy rules is 27, which are described by 27 integers in [1,3].1 to 3 denotes the three fuzzy sets of K. Position encoding of each particle is depicted in Fig.5. The front 16 dimensions is the MF parameters to be optimized, encoding in real number; and the rear 27 dimensions are the consequents of rules, encoding integers. The velocity of particle has the same dimensions as the position, but it encodes in real. Besides the position and the velocity of particles, the evaluate index, which is called fitness, should also be defined. The article selects the braking energy regenerative efficiency as fitness.

Due to the existence of ω, c_1 and c_2, although encoding of position or velocity in current iteration are integers, the next position or velocity may be real number. Therefore, the new real number position should take integer operation to avoid the infeasible solutions. $Round(x)$ operation is to implement the operation. It is defined as follows:

$$Round(x) = \begin{cases} floor(x) & x - floor(x) < 0.5 \\ floor(x) + 1 & x - floor(x) \geq 0.5 \end{cases}$$

Where, $floor(x)$ denotes integer operation.

In summary, the process for fuzzy controller design based PSO algorithm is as follow:

Step 1: Encode the MF parameters of inputs and outputs and consequents of rules as the searching space of PSO as Fig.4 presents, set the search range and the maximum velocity of particle.

Step 2: Initialize the speed and position of each particle. The MFs and rules we set in Table1 are added to the population as experienced particles.

Step 3: Update the speed and position of each particle.Then, take integer operation for particles` rules and positions using $Round(x)$ operation.

Step 4: Decode each particle, and output the results to fuzzy controller. Determine whether to update P_{id} and P_{gd} according to the fitness.

Step 5: Repeat Step 3 and Step 4 until a stop criterion is satisfied or a predefined number of iterations are completed. Then decode P_{gd} as the optimum parameters and rules of fuzzy controller.

Results

In order to validate the performance of the regenerative braking force distribution fuzzy controller based on PSO, compared to traditional fuzzy controller, simulations are taken on the typical drive cycle UDDS using ADVISOR software.

Fig.4.5.6 show the SOC trend and the torque using different algorithm during the specific cycle. From the Fig.4, we can see that the SOC trend using PSO-fuzzy algorithm decreases slower than that of traditional fuzzy algorithm. From Fig.5 and Fig.6 show that, in the first 200 seconds of simulation, although the motor torque is small, the braking energy can still be regenerated effectively; compared with traditional fuzzy algorithm, the vast majority of braking torque are ascending, in PSO algorithm; thus, the braking energy regeneration efficiency is improved obviously. From TABLE 3, we can see that the vehicle adopting traditional fuzzy controller has regenerative braking energy of 565J, while the one using the PSO-fuzzy controller is 687J, that is, adopting the latter controller, the value of braking energy regeneration efficiency is improved by 7.19 percent.

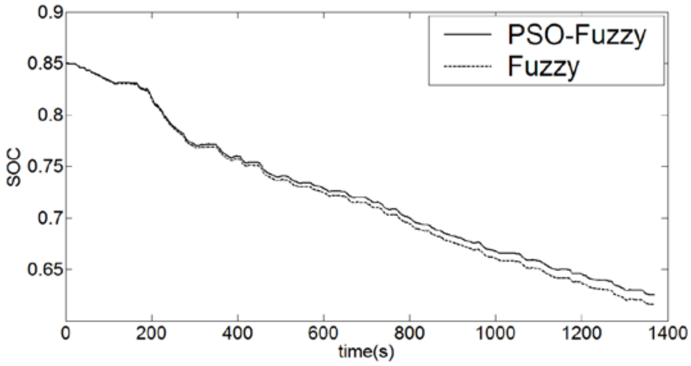

Fig.4 Diagram of SOC trend

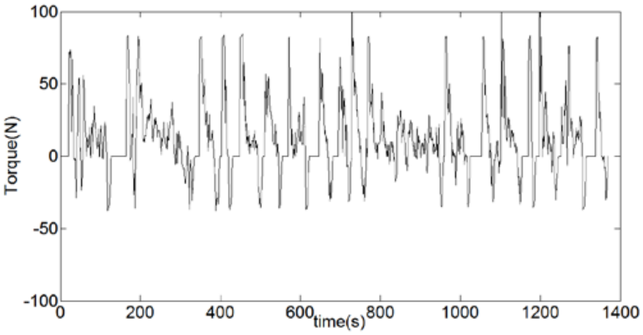

Fig.5 Diagram of motor torque using fuzzy controller

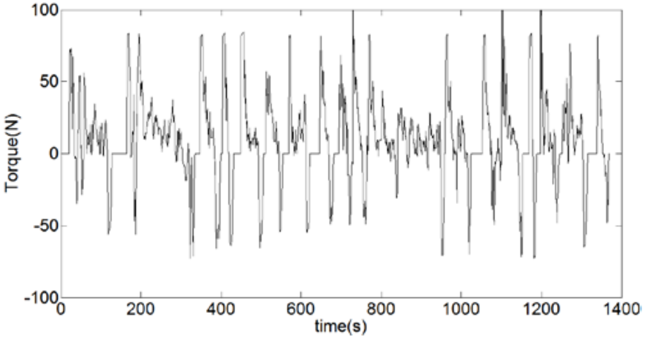

Fig.6 Diagram of motor torque using PSO-fuzzy controller

Table 3 Values comparison using two controllers

Controller type	Regeneration energy	Braking energy	Regeneration efficiency
Fuzzy controller	565 J	1697 J	33.29 percent
PSO-fuzzy controller	687 J	1697 J	40.48 percent

Conclusion

Regenerating braking energy under the conditions of the braking stability, according to this principle, this article uses PSO to optimize the MFs and rules of the regenerative braking force distribution fuzzy controller. Simulations are taken on the UDDS drive cycle using ADVISOR software, comparing with traditional fuzzy controller, the results show the braking energy regeneration efficiency is improved obviously using PSO algorithm.

Acknowledgements

This research was financially supported by the key technology of pure EV motor control and drive system research and design(cstc2014yykfA4004).

References

[1] Liu Hui, Wang Weida, He Jiao. Modeling and Simulation of the Regenerative Braking System in a HEV Based on Fuzzy Control[J]. Automotive Engineering, 2012,34(1):51-56.

[2] Zhang Zijian, Xu Guoqing, Li Weimin. Regenerative Braking for Electric Vehicle Based on fuzzy logic control strategy [C]. Mechanical and Electronics Engineering(ICMEE), 2010 2nd International Conference on, Vol 1. IEEE, 2010:319-323.

[3] Wu Qidi, Wang Lei. Intelligent Particle Swarm Optimization Algorithm Research and Application[M]. Nan Jing: Jiangsu Education Publishing House,2005:106-113.

[4] Li Li, Niu Ben. Particle Swarm Optimization Algorithm[M]. Bei Jing: Metallurgical Industry Press,2010:25-33.

[5] Shi Yuhui, Russell C.Eberhart. Parameter Selection in Particle Swarm Optimization Algorithm[C]. The 7th Annual Conference on Evolutionary Programming, San Diego, USA, 1998.