EGR Intelligent Control of Diesel Engine Based on Deep Reinforcement Learning

ChenGuang Lai, ChaoBing Wu, SiZheng Wang, JiaXi Li, and Bo Hu(✉)

Chongqing University of Technology, Chongqing, China
b.hu@cqut.edu.cn

Abstract. Intelligent Connected Vehicle (ICV), as a revolutionary technology for automobiles, is rapidly developing and changing the way people travel. However, the current smart cars lack the intelligent control of the powertrain, and even if the network connection is completed, the power, economy and emissions cannot be greatly improved. State-of-the-art deep reinforcement learning algorithms, whose agents continuously interact with the model, employ an end-to-end control strategy. The deep learning neural network is used to fit the mapping relationship between the state and the action, and the action of the agent is evaluated by the reinforcement learning reward function, and iteratively learns the control strategy that meets the goal. This paper adopts a new EGR control method based on deep reinforcement learning, and compares it with the traditional PID control method to verify whether the method is feasible and provide a reference for the intelligent control of the engine.

Keywords: Deep Reinforcement Learning (DRL) · Exhaust Gas Recycling (EGR) · Mean Value Model

1 Introduction

With the implementation of the latest automobile emission regulations and the improvement of NOx standards, it has reached the technical bottleneck to optimize automobile emissions by general means. Now the engine has to optimize automobile emissions through EGR system. EGR (exhaust gas recirculation) system is an exhaust gas recirculation system that introduces part of the exhaust gas into the intake manifold and mix fresh air into the cylinder. Due to the additional exhaust gas get into the cylinder, the increase of combustion temperature can be effectively restrained. The increase of inert gas that prolongs the ignition delay period and slows down the combustion velocity leads to effectively inhibit the formation of nitrogen oxides. As a result, an accurate controlletion of the EGR valve give appropriate amount and suitable time of exhaust gas into the cylinder. The EGRrate, the ratio of exhaust gas flow to the total flow into the cylinder, is a parameter indicates the amount of exhaust gas. At present, centralized electronic control technology is widely used in automobile, which can realize precise control of EGR valve learning and intelligent control. At the same time, it may be a very good choice to meet the demand for simplicity and robustness inindustrial [3–5].

L. Yan and J. Na (Eds.): ICFPMCE 2022, AHE 10, pp. 151–161, 2023.
https://doi.org/10.2991/978-94-6463-022-0_14

Meanwhile, with the long research time, high reliability and wide application range [1, 2], PID is often used as the controller in industry. For diesel engine, the interveneces will be araised between VGT and EGR. The pressure of exhaust manifold changed by VGT will increase the difficulty of EGR control. The traditional PID control method with poor versatility and high cost is difficult to tune parameters. At present, expert PID control, fuzzy-PID control and PID control based on neural network can achieve better optimization effect. However, these control methods need to learn expert knowledge, build fuzzy control decision table and tune complex neural network parameters, which will become very complex when applied to EGR, and can not meet the requirements of industrial application.

As an important development direction of artificial intelligence, machine learning has made great progress in games, go, image recognition and other fields. Deep reinforcement learning is considered to be one of the best algorithms to artificial intelligence, and has a good development in various industries. Reinforcement learning is mainly composed of agent, environment, state, action and reward. Agent obtains feedback by interacting with environment and makes an action to environment according to certain strategy; Through reinforcement learning, agents can get the information of their state and the actions they should take to get the maximum reward. The interactive method of agent is like human, and it can be considered that reinforcement learning is a general learning framework to solve AI problems.

As an important method of machine learning, deep reinforcement learning (DRL) is regarded as an important means to realize artificial intelligence; DRL has the advantages of supervised learning and unsupervised learning, and has great superiorities in dealing with complex decision-making problems. DRL can be divided into on policy and off policy on the basis of policy. The most representative algorithm of on policy is *Sarsa* and $Sarsa(\lambda)$; Q-learning algorithm is the most breakthrough algorithm of off policy. It is the first method to defeat the top chess players in the field of artificial intelligence [6, 7]. The Deep Q Network (DQN) strategy adopted on Alpha-Go [8] and Alpha-Zero [9] is the first computer program to beat a human professional Go player. The utility of deep network learning replace Q-table accurately inhance the adaptive capacity of RL, so that it can realize the high-dimensional state and action space. The DDPG Strategy proposed by Lillicrap et al. [10] is a deep reinforcement learning algorithm combining DQN algorithm and Actor-Critic algorithm, which uses deep neural network as the approximator. It solves the problem that the DQN algorithm can only output discrete actions, which can not be applied to most real control tasks, and the Actor-Critic algorithm is difficult to converge and unstable.

In the following papers, DDPG algorithm is used to realize the continuous control of EGR solenoid valve of diesel engine. The structure of the thesis is as follow: The second section proposed the optimal control of EGR rate based on deep reinforcement learning; The third section present comparison and discussion of the results of deep reinforcement learning and PID control; The fourth part give the conclusion of the research.

2 Method

2.1 Deep Reinforcement Learning Algorithm

Model free deep reinforcement learning is a self-learning and self optimizing decision-making method. It is different from most other control algorithms in that it learns an optimal strategy through the interaction between agent and environment. Figure 1 is a process of interaction between agent and environment; The agent first observes the environment to get a state(s), according to which the agent makes an action on the environment, then observes the state(s')after the environment changes, and gets a reward value(r) by evaluating the changed state, which can evaluate the agent's action; This process can be described by Markov chain (MDP) [11, 12]. The essence of reinforcement learning is to describe the problem as a Markov decision process and find an optimal strategy. The so-called strategy is the mapping from state to action, expressed by $\pi(a|s) = p[A_t = a|S_t = s]$. The reinforcement learning uses a reward function to represent the return value of a specific time step:

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{1}$$

G_t represents the total reward value of a certain time steps, and R represents an immediate reward in that time step, γ represents the discount factor of a reward value (the further away from the current time step, the smaller the impact on the total reward value, so its index is increasing).

DDPG algorithm combines the the advantages of actor-critic framework and and DQN, and its core root formula is the well-known Bellman recursive formula. The formula is as follow:

$$\begin{aligned} Q_\pi(s, a) &= E_n[G_0|S_0 = s, A_0 = a] \\ &= E[\int_{T=0}^{0} r^t R_{t+1}||S_0 = s, A_0 = a] \end{aligned} \tag{2}$$

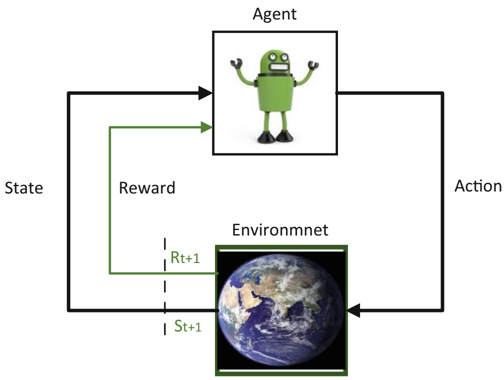

Fig. 1. Basic framework of reinforcement learning

Although the DQN can deal with the state input of high-dimensional, its action is still discrete and can not deal with the continuously decision-making problem; The DDPG strategy uses deep neural networks as approximators to effectively combine deep learning and deterministic strategy gradient algorithms. It can cope with high-dimensional inputs, achieve end-to-end control, output continuous actions, and thus can be applied to more complex situations with large state spaces and continuous action spaces. In detail, DDPG uses an actor network to tune the parameter θ^{μ} for the policy function, that is, decide the optimal action for a given state. A critic is used for evaluating the policy function estimated by the actor according to the temporal TD error (see Fig. 2) [13–15].

The engine speed, actual EGR rate, target EGR rate and current EGR valve opening are taken as the four-dimensional state space of DDPG algorithm; Continuous control of EGR valve opening as an action space. Immediate reward is very important in RL algorithm, which directly affects the convergence curve; In some cases, even if you fine

Table 1. Pseudo-code of the Deep deterministic Policy gradient (DDPG) algorithm.

DDPG algorithm:
Randomly initialize critic network $Q(s, a\|\theta^{Q})$ and actor $\mu(s\|\theta^{\mu})$ with weights θ^{Q} and θ^{μ}
Initialize target network Q and μ with weights $\theta^{Q'} \leftarrow \theta^{Q}, \theta^{\mu'} \leftarrow \theta^{\mu}$
for episode = 1, M do
Initialize a random process N for action exploration
Receive initial observation state s_1
for t = 1, T do
Select action $a = \mu(s_t\|\theta^{\mu}) + N_t$ according to the current policy and exploration noise
Execute action a_t and observe reward r_t and observe new state s_{t+1}
Store transition $(s_t, a_t, r_t, s_{t+1}))$ in R
Sample a random minibatch of N transitions (s_t, a_t, r_t, s_{t+1}) from R
Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}\|\theta^{\mu'})\|\theta^{Q'})$
Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i\|\theta^{Q}))^2$
Update the actor policy using the sampled gradient:
$\nabla_{\theta^{\mu}} J \approx \frac{1}{N}\sum_i \nabla_a Q(s, a\|\theta^{Q}\|a = s_i, a = \mu(s_i) \cdot \nabla_{\theta^{\mu}}\mu(s\|\theta^{\mu})\|_{s_i}$
Update the target networks:
$\theta^{Q'} \leftarrow \tau\theta^{Q} + (1 - \tau)\theta^{Q'}$
$\theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1 - \tau)\theta^{\mu'}$
end for
end for

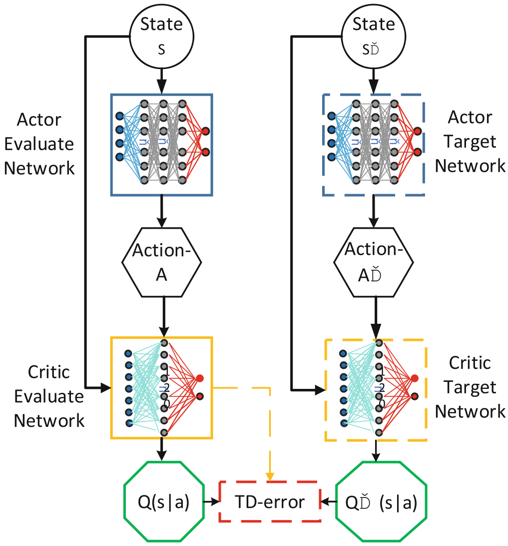

Fig. 2. Actor-critic architecture.

tune the immediate reward, the convergence results can change dramatically. Therefore, the immediate reward should be defined according to the optimization goal, and the error and action change rate between the current EGR rate and the target EGR rate are regarded as the immediate reward. The formula is as follow:

$$r_t = e^{-\frac{[0.95*|e(t)|+0.05*|I_t|]^2}{2}} - 1 \tag{3}$$

r_t represents the immediate reward, $e(t)$ and I_t represents the error and action change rate between the current EGR rate and the target EGR rate respectively. Table 1 shows some super-parameters used in DDPG. Actor uses two-layer deep neural network with 300 neurons in each layer. Critical uses one-layer neural network with 300 neurons (Table 2).

2.2 Co-simulation Platform

The simulation is conducted on a co-simulation platform. The control algorithm is written through Python language, and the model is established in GT-SUITE. There is no direct communication between the two, so it is necessary to use MATLAB/Simulink to realize real-time data transmission between GT-SUITE and python. The structure of the system is shown in Fig. 3.

2.3 Mean Value Model

The engine modeling and simulation analysis is the basis of engine control system design. Engine model plays an important role in the whole process. The correct simulation result is based on a model which has correct parameters and compete structure. Mean value

Table 2. DDPG parameters.

Parameters	Value
Learning rate for actor	0.0001
Learning rate for critic	0.0001
Reward discount factor γ	0.9
Soft replacement factor τ	0.01
Replay memory size	100000
Mini-batch size	128
Action randomness decay	0.999995
Initial exploration	10

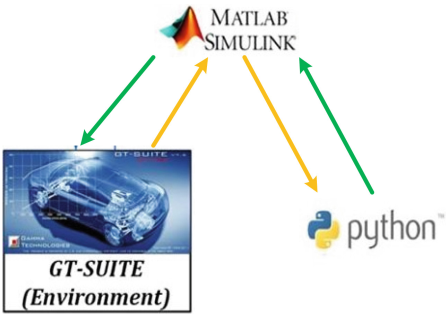

Fig. 3. Co-simulation platform

engine model has been widely used in engine control research. It combines the calculation advantages of accuracy in detail model and fast in Fast Running Model (FRM) [16].

Mean value engine models are useful for certain types of modeling where simulation speed is of primary importance, the details of wave dynamics are not critical, and bulk fluid flow is still important. A mean value engine model essentially contains a map based cylinder model that is computationally faster than a regular (detailed) cylinder. The simulation speed can be increased further by combining multiple detailed cylinders into a single mean value cylinder.

The basis of a mean value engine model is the mean value cylinder, which is simply a map based cylinder. The three maps that determine the mean value cylinder performance are volumetric efficiency, indicated mean effective pressure (IMEP), and exhaust gas temperature. Each of these three quantities is an input to the mean value cylinder and is imposed by the cylinder during a simulation. So to build a realistic mean value model, it is necessary to define each of these three quantities as a function of other variables [17, 18].

Volumetric Efficiency should nearly always be a function of engine speed (unless a constant speed engine) and intake manifold pressure. Other variables to consider based on the specific engine details and intended purpose of the model are intake manifold

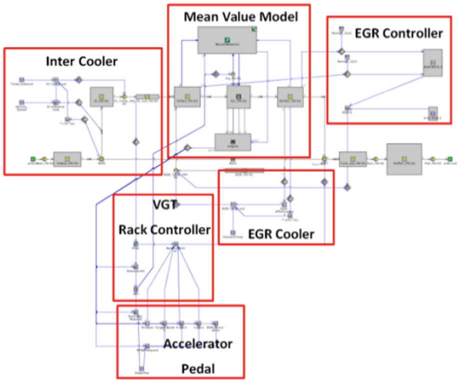

Fig. 4. Reinforcement learning control EGR average model

temperature, exhaust manifold pressure, valve event timing and profile, and burned gas fraction in intake manifold (for EGR engines). IMEP and Exhaust Gas Temperature will generally be dependent on the same variables as they are both related to distribution of fuel energy. They also generally will be dependent on the trapped air mass in the cylinder and should therefore contain at a minimum the same dependencies as volumetric efficiency. The mean value engine model established by fitting these three parameters greatly improves the simulation speed and enhances the efficiency of research without reducing accuracy [19].

2.3.1 The Composition of Mean Value Engine Model

As shown in Fig. 4 is the composition of the whole engine model which EGR is controlled by reinforcement learning. It's mainly composed with six parts, and they are mean value engine model, intercooler, EGR controller, VGT position controller, accelerator pedal and EGR cooling module. The engine model is a 4-cylinder 3-L turbocharged direct injection diesel engine. The EGR controller uses reinforcement learning method to control the EGR valve to achieve the EGR control of the whole mean value engine model.

Volumetric efficiency, IMEP and exhaust gas temperature define the cylinder of mean value engine model. These three quantities are calculated by neural networks which depend on seven input variables (engine speed, intake manifold pressure and temperature, exhaust manifold pressure, fuel rate, injection timing and EGR fraction). In addition, a neural network with seven variables is used to calculate the FMEP of crankshaft.

The actual EGR rate is calculated with the mass fraction of CO2 sensed by the sensors called CO_2_INLET and CO_2_OUTLET in the intake manifold, and the value will be transmitted to the EGR-controller. The EGR-controller controls EGR by adjusting EGR valve based on the target EGR rate in time-EGR part. The actual EGR-rate calculate formula is as follow:

$$\eta_{(EGR-rate)} = \frac{CO_2\_INLET}{CO_2\_OUTLET} \times 100\% \quad (4)$$

3 Results and Discussion

In order to verify the effect of reinforcement learning control, the FTP-72 (Federal Test Procedure) of the United States shown in Fig. 5 is selected as the working condition. The working condition simulates a 12.07 km urban line, and often carries out rapid acceleration and deceleration, with the highest speed of 91.25 km/h and the average speed of 31.5 km/h. The reason why chose this working condition is that it can represent the real engine working condition with large delay, strong coupling and nonlinearity. If it can train a better control strategy in this complex environment, it can perform better in other stable conditions such as Europe NEDC.

The target EGR rate of the whole working condition can be calculated according to the vehicle speed, engine speed, pedal position and our mean value engine model; The whole working condition has 1372 s, and the DRL algorithm needs some training to determine some parameters, which greatly increases a time cost; In this regard, we choose a period of time with rapid and complex changes in the middle condition for training, and a training condition between 800 s to 850 s will be choosed. As shown in Fig. 6, the red line is a target EGR rate calculated according to the working condition

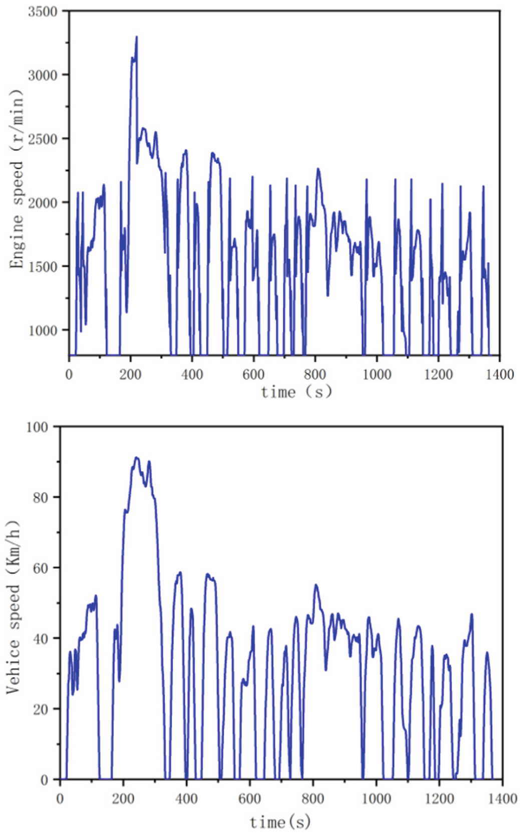

Fig. 5. FTP72 engine speed and vehicle speed diagram

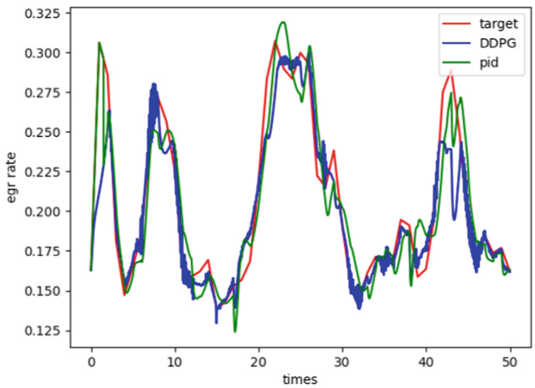

Fig. 6. DRL and PID Control Result

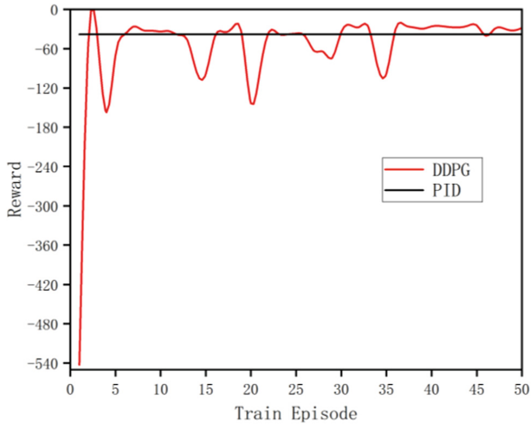

Fig. 7. DDPG Control Reward

data in the engine model; The green line is the result of a PID controller which use a PI control, and it is a good result obtained after continuous optimization and adjustment of parameters. The blue line is an effect of DRL training. Only 50 rounds have been trained this time and both DRL and PID are not well controlled in local position. The reason is the controller has a certain delay, which can not be solved by the control itself; PID has a certain overshoot in some local conditions, while DRL is relatively better; This indicates that deep reinforcement learning is feasible in engine control, and can provide reference for improving engine control.

Reward is an important value to evaluate whether the algorithm has gotten convergence. Reward is a Gaussian function about action and state; Reference to the label of supervised learning is an important method to evaluate the results of DRL training. Showing in Fig. 7, the agent is constantly trying and making mistakes at start, and the reward value is always very low. Through continuous learning, the agent converges in

37 rounds, but it fluctuates in a small range after convergence. This is because the output of the neural network has randomness, but it has little influence on the final control.

4 Conclusions

A new control method is proposed in this paper, which applies DRL algorithm to the control of engine bottom actuator to realize end-to-end control; Compared with the mature PID control, the DRL control is better in local conditions, which is mainly owing to the characteristics of self-learning and self-adaptive of DRL; Therefore, the application of DRL in engine control is feasible, and the control effect can be improved by continuously optimizing the algorithm structure and parameters, so as to provide a feasible method for engine control.

References

1. Yokomura, H., et al.: Transient EGR control for a turbocharged heavy duty diesel engine. No. 2004-01-0120. SAE Technical Paper (2004)
2. Ammann, M., et al.: Model-based control of the VGT and EGR in a turbocharged common-rail diesel engine: theory and passenger car implementation. SAE Trans. 527–538 (2003)
3. Amstutz, A., Del Re, L.R.: EGO sensor based robust output control of EGR in diesel engines. IEEE Trans. Control Syst. Technol. **3**(1), 39–48 (1995)
4. Pradeep, V., Sharma, R.P.: Use of HOT EGR for NOx control in a compression ignition engine fuelled with bio-diesel from Jatropha oil. Renew. Energy **32**(7), 1136–1154 (2007)
5. Martinez-Frias, J., et al.: Equivalence ratio-EGR control of HCCI engine operation and the potential for transition to spark-ignited operation. No. 2001-01-3613. SAE Technical paper (2001)
6. Liu, T., Zou, Y., Liu, D., Sun, F.: Reinforcement learning-based energy management strategy for a hybrid electric tracked vehicle. Energies **8**, 7243–7260 (2015)
7. Cao, H., Yu, T., Zhang, X., Yang, B., Wu, Y.: Reactive power optimization of large-scale power systems: a transfer Bees optimizer application. Processes **7**, 321 (2019)
8. Silver, D., et al.: Mastering the game of go with deep neural networks and tree search. Nature **529**, 484–489 (2016)
9. Silver, D., et al.: Mastering the game of go without human knowledge. Nature **550**, 354–359 (2017)
10. Howard, R.A.: Dynamic Programming and Markov Process. The MIT Press, Cambridge, MA, USA (1966)
11. Wang, Y., Velswamy, K., Huang, B.: A long-short term memory recurrent neural network based reinforcement learning controller for office heating ventilation and air conditioning systems. Processes **5**, 46 (2017)
12. Proceedings of the International Conference on Learning Representations, Continuous control with deep reinforcement learning, San Juan, PR, USA, 2–4 May 2016
13. Hu, B., et al.: Energy management strategy of hybrid vehicle based on parallel deep reinforcement learning. J. Chongqing Univ. Technol. (Nat. Sci.) **34**(9) (2020)
14. Hu, B., et al.: A hybrid end-to-end control strategy combining dueling deep q-network and PID for transient boost control of a diesel engine with variable geometry turbocharger and cooled EGR. Energies **12**(19), 3739 (2019)
15. Hu, B., et al.: Reinforcement learning approach to design practical adaptive control for a small-scale intelligent vehicle. Symmetry **11**(9), 1139 (2019)

16. Silver, D., et al.: Mastering the game of go with deep neural networks and tree search. Nature **529**(7587), 484–489 (2016). https://doi.org/10.1038/nature16961
17. Hu, B., Jiaxi, L.: An edge computing framework for powertrain control system optimization of intelligent and connected vehicles based on curiosity-driven deep reinforcement learning. IEEE Trans. Ind. Electron. (2020)
18. Hu, B., Yang, J., Li, J., Li, S., Bai, H.: Intelligent control strategy for transient response of a variable geometry turbocharger system based on deep reinforcement learning. Processes **7**(9) (2019)
19. Nair, A., et al.: Massively parallel methods for deep reinforcement learning (2015). arXiv preprint arXiv:1507.04296